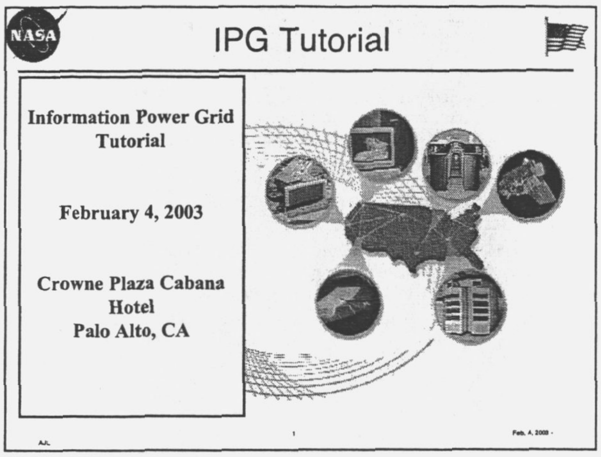


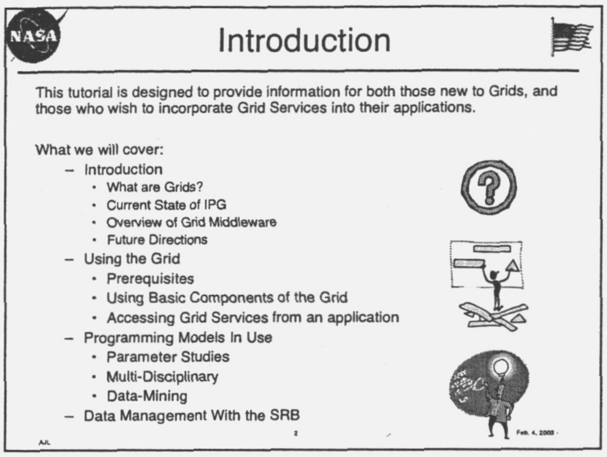
IPG Tutorial

Information Power Grid Tutorial

February 4, 2003

Crowne Plaza Cabana Hotel
Palo Alto, CA

AJL 1 Feb. 4, 2003 -



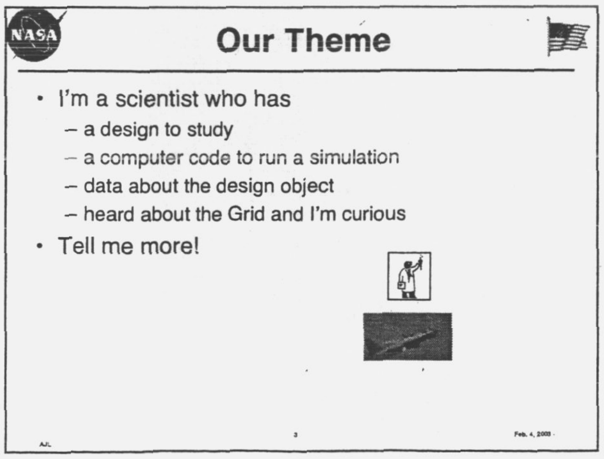
Introduction

This tutorial is designed to provide information for both those new to Grids, and those who wish to incorporate Grid Services into their applications.

What we will cover:

- Introduction
  - What are Grids?
  - Current State of IPG
  - Overview of Grid Middleware
  - Future Directions
- Using the Grid
  - Prerequisites
  - Using Basic Components of the Grid
  - Accessing Grid Services from an application
- Programming Models In Use
  - Parameter Studies
  - Multi-Disciplinary
  - Data-Mining
- Data Management With the SRB

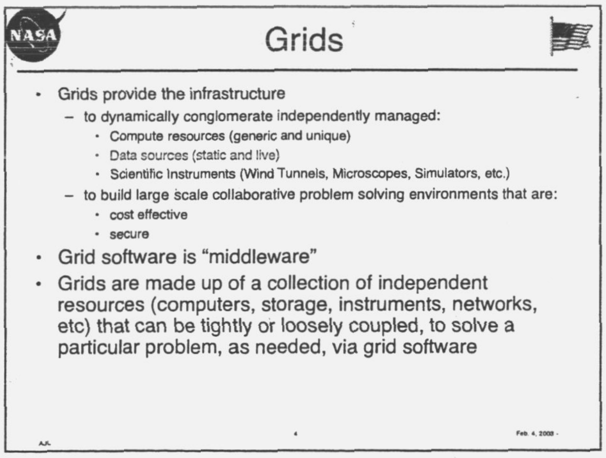
AJL 2 Feb. 4, 2003 -



Our Theme

- I'm a scientist who has
  - a design to study
  - a computer code to run a simulation
  - data about the design object
  - heard about the Grid and I'm curious
- Tell me more!

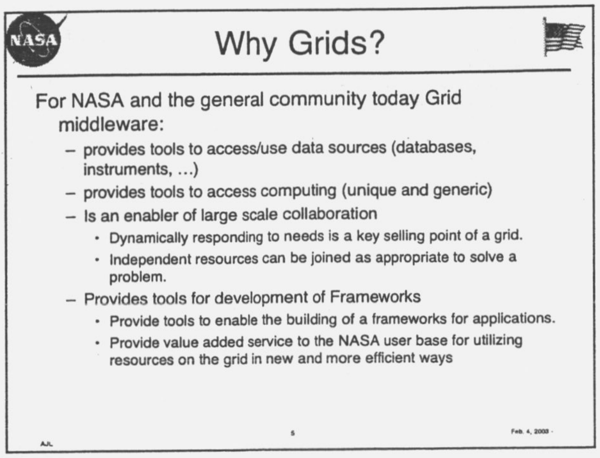
AJL 3 Feb. 4, 2003 -



Grids

- Grids provide the infrastructure
  - to dynamically conglomerate independently managed:
    - Compute resources (generic and unique)
    - Data sources (static and live)
    - Scientific Instruments (Wind Tunnels, Microscopes, Simulators, etc.)
  - to build large scale collaborative problem solving environments that are:
    - cost effective
    - secure
- Grid software is "middleware"
- Grids are made up of a collection of independent resources (computers, storage, instruments, networks, etc) that can be tightly or loosely coupled, to solve a particular problem, as needed, via grid software

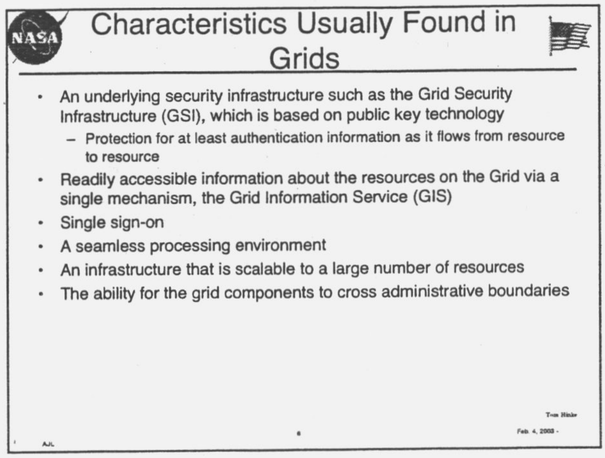
AJL 4 Feb. 4, 2003 -



Why Grids?

For NASA and the general community today Grid middleware:

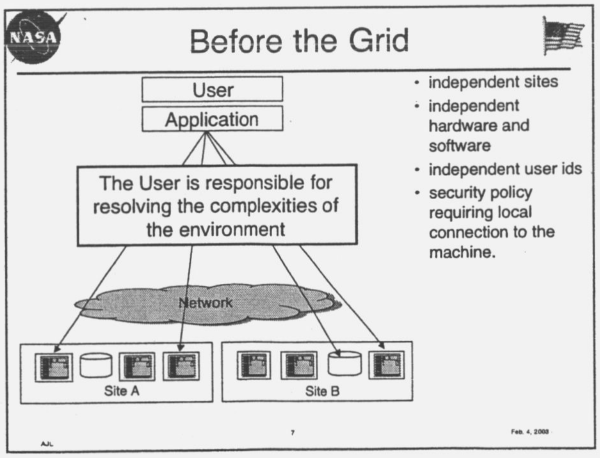
- provides tools to access/use data sources (databases, instruments, …)
- provides tools to access computing (unique and generic)
- Is an enabler of large scale collaboration
  - Dynamically responding to needs is a key selling point of a grid.
  - Independent resources can be joined as appropriate to solve a problem.
- Provides tools for development of Frameworks
  - Provide tools to enable the building of a frameworks for applications.
  - Provide value added service to the NASA user base for utilizing resources on the grid in new and more efficient ways



Characteristics Usually Found in Grids

- An underlying security infrastructure such as the Grid Security Infrastructure (GSI), which is based on public key technology
  - Protection for at least authentication information as it flows from resource to resource
- Readily accessible information about the resources on the Grid via a single mechanism, the Grid Information Service (GIS)
- Single sign-on
- A seamless processing environment
- An infrastructure that is scalable to a large number of resources
- The ability for the grid components to cross administrative boundaries

Tom Hinke



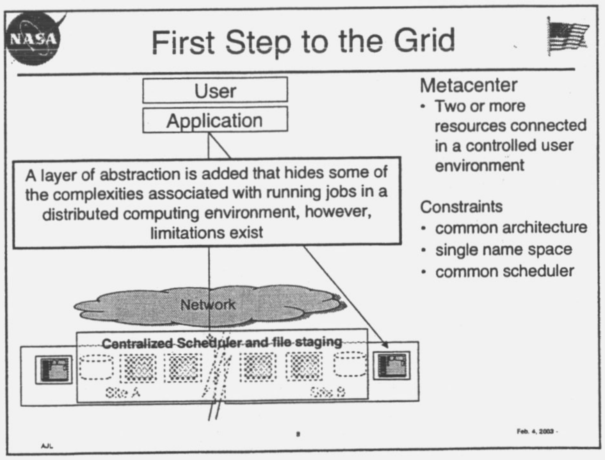
Before the Grid

User

Application

The User is responsible for resolving the complexities of the environment

- independent sites
- independent hardware and software
- independent user ids
- security policy requiring local connection to the machine.



First Step to the Grid

User

Application

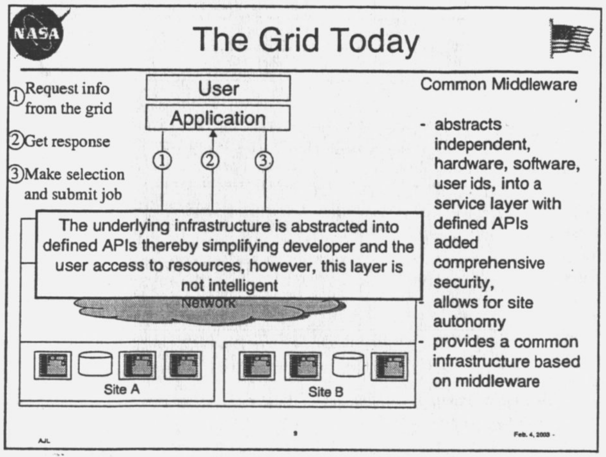
A layer of abstraction is added that hides some of the complexities associated with running jobs in a distributed computing environment, however, limitations exist

Metacenter

- Two or more resources connected in a controlled user environment

Constraints

- common architecture
- single name space
- common scheduler



The Grid Today

①Request info from the grid

②Get response

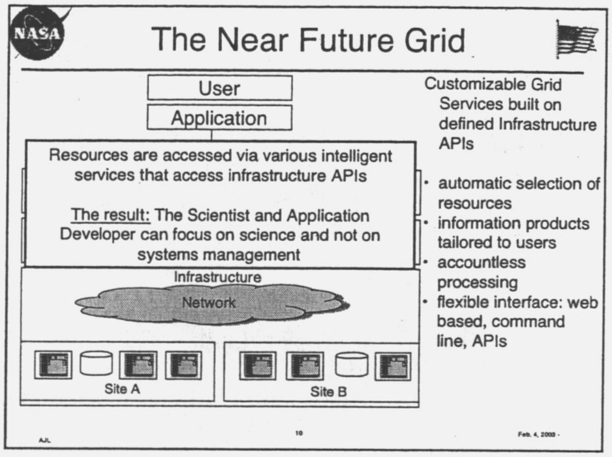
③Make selection and submit job

The underlying infrastructure is abstracted into defined APIs thereby simplifying developer and the user access to resources, however, this layer is not intelligent

Common Middleware

- abstracts independent, hardware, software, user ids, into a service layer with defined APIs
- added comprehensive security,
- allows for site autonomy
- provides a common infrastructure based on middleware

AJL 9 Feb. 4, 2003 -



The Near Future Grid

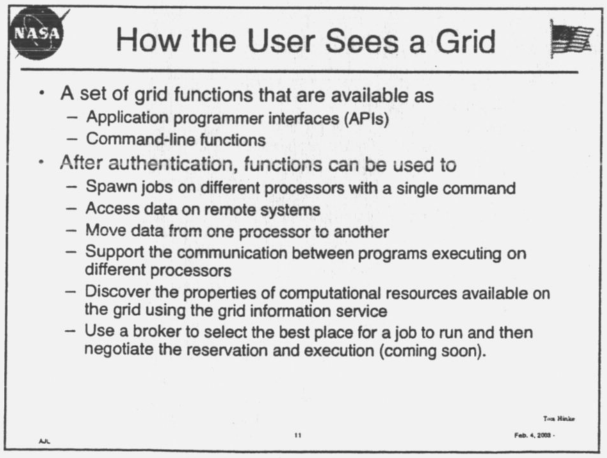
Resources are accessed via various intelligent services that access infrastructure APIs

The result: The Scientist and Application Developer can focus on science and not on systems management

Customizable Grid Services built on defined Infrastructure APIs

- automatic selection of resources
- information products tailored to users
- accountless processing
- flexible interface: web based, command line, APIs

AJL 10 Feb. 4, 2003 -

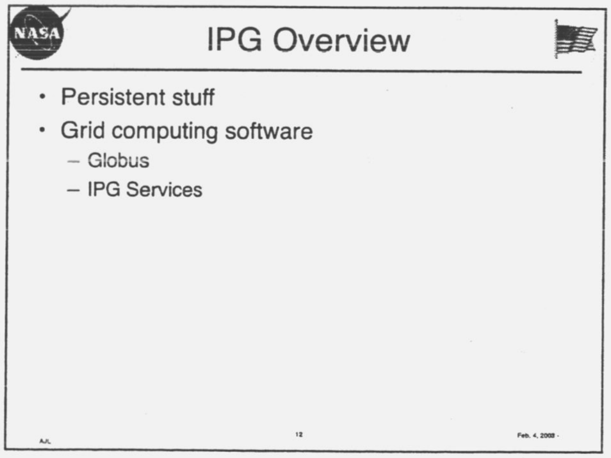


How the User Sees a Grid

- A set of grid functions that are available as
  - Application programmer interfaces (APIs)
  - Command-line functions
- After authentication, functions can be used to
  - Spawn jobs on different processors with a single command
  - Access data on remote systems
  - Move data from one processor to another
  - Support the communication between programs executing on different processors
  - Discover the properties of computational resources available on the grid using the grid information service
  - Use a broker to select the best place for a job to run and then negotiate the reservation and execution (coming soon).

Tom Hinke

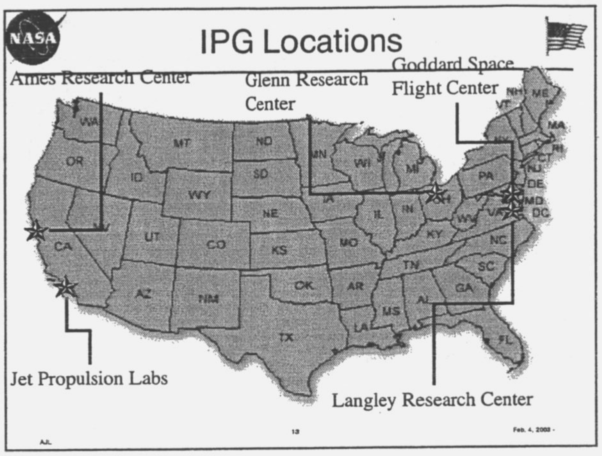
AJL 11 Feb. 4, 2003 -



IPG Overview

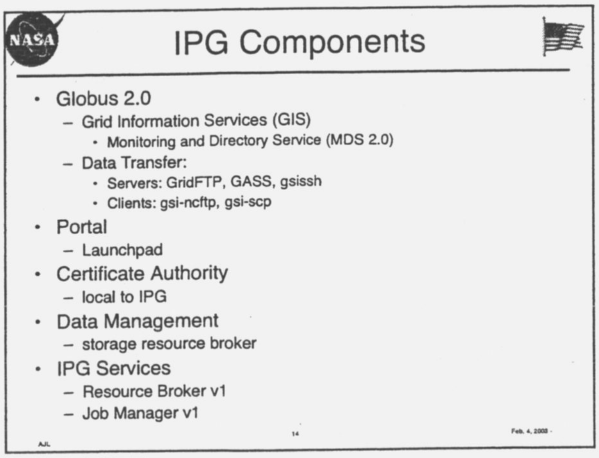
- Persistent stuff
- Grid computing software
  - Globus
  - IPG Services

AJL 12 Feb. 4, 2003 -



IPG Locations

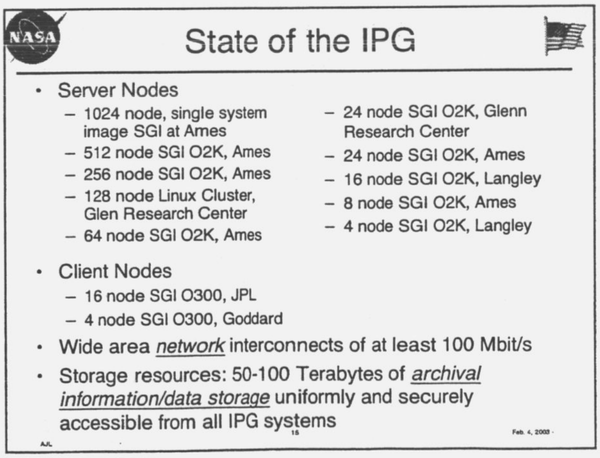
AJL 13 Feb. 4, 2003 -



IPG Components

- Globus 2.0
  - Grid Information Services (GIS)
    - Monitoring and Directory Service (MDS 2.0)
  - Data Transfer:
    - Servers: GridFTP, GASS, gsissh
    - Clients: gsi-ncftp, gsi-scp
- Portal
  - Launchpad
- Certificate Authority
  - local to IPG
- Data Management
  - storage resource broker
- IPG Services
  - Resource Broker v1
  - Job Manager v1

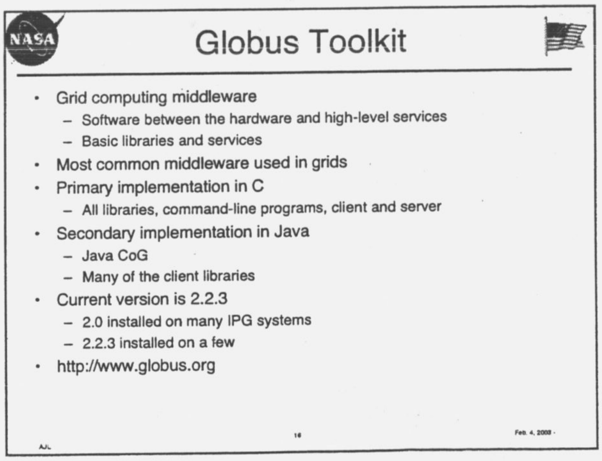
AJL 14 Feb. 4, 2003 -



State of the IPG

- Server Nodes
  - 1024 node, single system image SGI at Ames
  - 512 node SGI O2K, Ames
  - 256 node SGI O2K, Ames
  - 128 node Linux Cluster, Glen Research Center
  - 64 node SGI O2K, Ames
  - 24 node SGI O2K, Glenn Research Center
  - 24 node SGI O2K, Ames
  - 16 node SGI O2K, Langley
  - 8 node SGI O2K, Ames
  - 4 node SGI O2K, Langley
- Client Nodes
  - 16 node SGI O300, JPL
  - 4 node SGI O300, Goddard
- Wide area *network* interconnects of at least 100 Mbit/s
- Storage resources: 50-100 Terabytes of *archival information/data storage* uniformly and securely accessible from all IPG systems

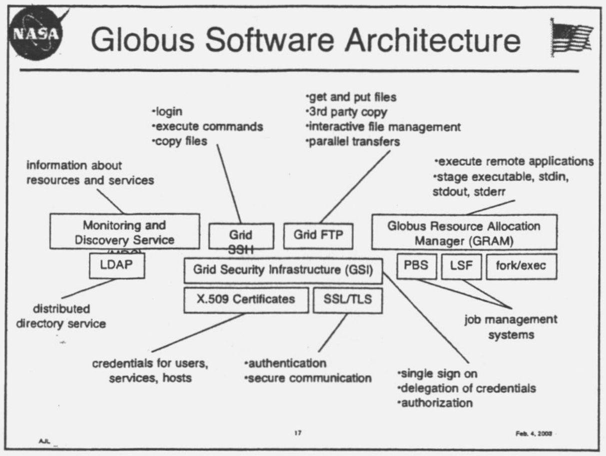
AJL 15 Feb. 4, 2003 -



Globus Toolkit

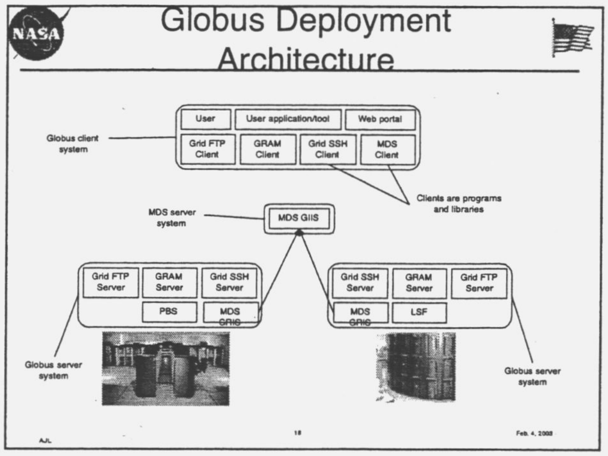
- Grid computing middleware
  - Software between the hardware and high-level services
  - Basic libraries and services
- Most common middleware used in grids
- Primary implementation in C
  - All libraries, command-line programs, client and server
- Secondary implementation in Java
  - Java CoG
  - Many of the client libraries
- Current version is 2.2.3
  - 2.0 installed on many IPG systems
  - 2.2.3 installed on a few
- http://www.globus.org

AJL 16 Feb. 4, 2003 -

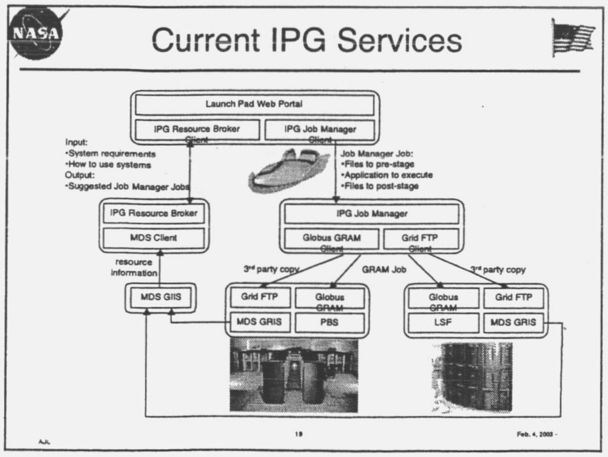


Globus Software Architecture

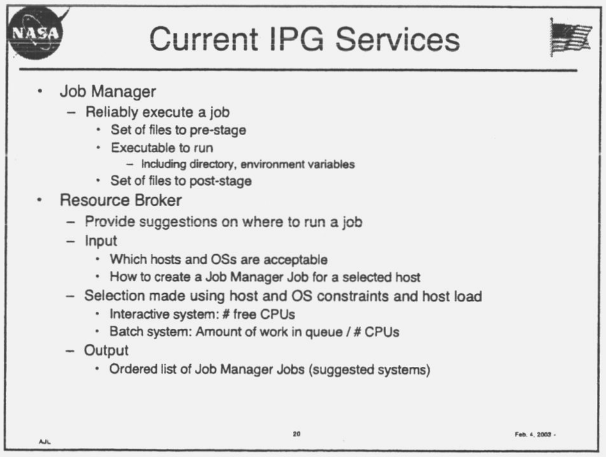
•login
•execute commands
•copy files
•get and put files
•3rd party copy
•interactive file management
•parallel transfers
information about resources and services
•execute remote applications
•stage executable, stdin, stdout, stderr
Monitoring and Discovery Service (MDS)
Grid SSH
Grid FTP
Globus Resource Allocation Manager (GRAM)
LDAP
Grid Security Infrastructure (GSI)
PBS
LSF
fork/exec
X.509 Certificates
SSL/TLS
distributed directory service
job management systems
credentials for users, services, hosts
•authentication
•secure communication
•single sign on
•delegation of credentials
•authorization



Globus Deployment Architecture

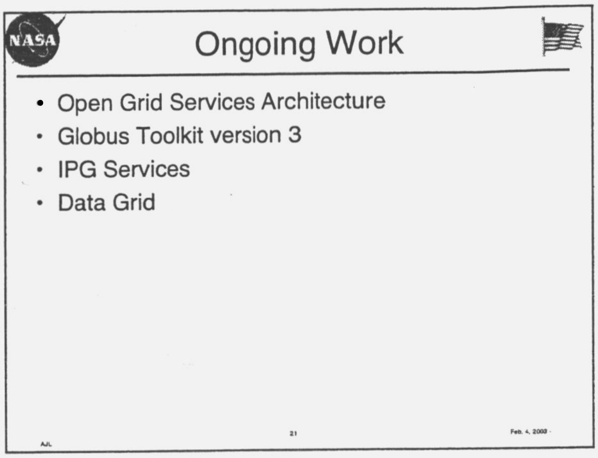


Current IPG Services



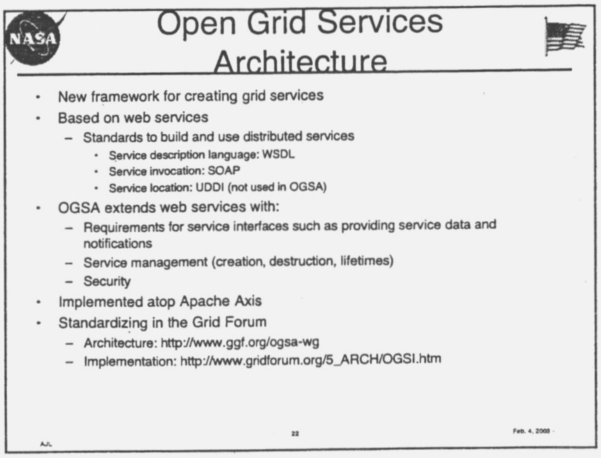
Current IPG Services

- Job Manager
  - Reliably execute a job
    - Set of files to pre-stage
    - Executable to run
      - Including directory, environment variables
    - Set of files to post-stage
- Resource Broker
  - Provide suggestions on where to run a job
  - Input
    - Which hosts and OSs are acceptable
    - How to create a Job Manager Job for a selected host
  - Selection made using host and OS constraints and host load
    - Interactive system: # free CPUs
    - Batch system: Amount of work in queue / # CPUs
  - Output
    - Ordered list of Job Manager Jobs (suggested systems)



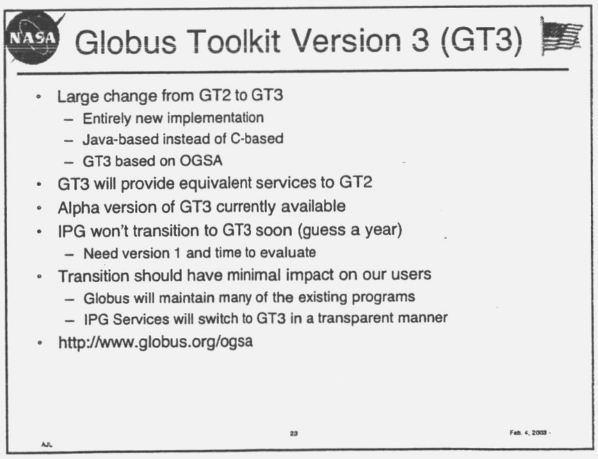
Ongoing Work

- Open Grid Services Architecture
- Globus Toolkit version 3
- IPG Services
- Data Grid



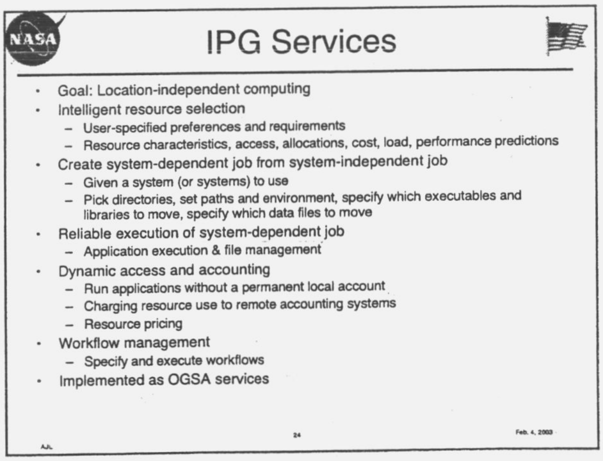
Open Grid Services Architecture

- New framework for creating grid services
- Based on web services
  - Standards to build and use distributed services
    - Service description language: WSDL
    - Service invocation: SOAP
    - Service location: UDDI (not used in OGSA)
- OGSA extends web services with:
  - Requirements for service interfaces such as providing service data and notifications
  - Service management (creation, destruction, lifetimes)
  - Security
- Implemented atop Apache Axis
- Standardizing in the Grid Forum
  - Architecture: http://www.ggf.org/ogsa-wg
  - Implementation: http://www.gridforum.org/5_ARCH/OGSI.htm



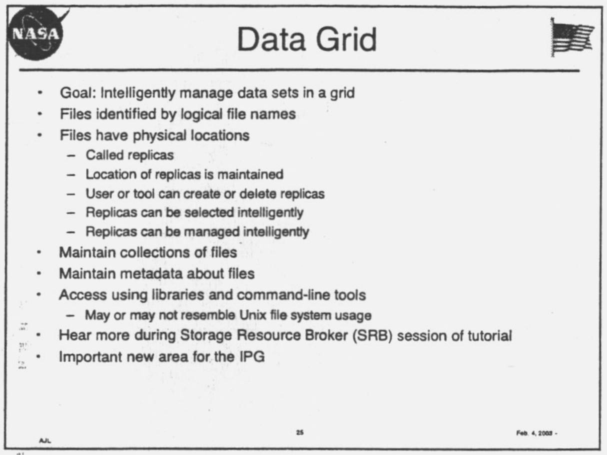
Globus Toolkit Version 3 (GT3)

- Large change from GT2 to GT3
  - Entirely new implementation
  - Java-based instead of C-based
  - GT3 based on OGSA
- GT3 will provide equivalent services to GT2
- Alpha version of GT3 currently available
- IPG won't transition to GT3 soon (guess a year)
  - Need version 1 and time to evaluate
- Transition should have minimal impact on our users
  - Globus will maintain many of the existing programs
  - IPG Services will switch to GT3 in a transparent manner
- http://www.globus.org/ogsa



IPG Services

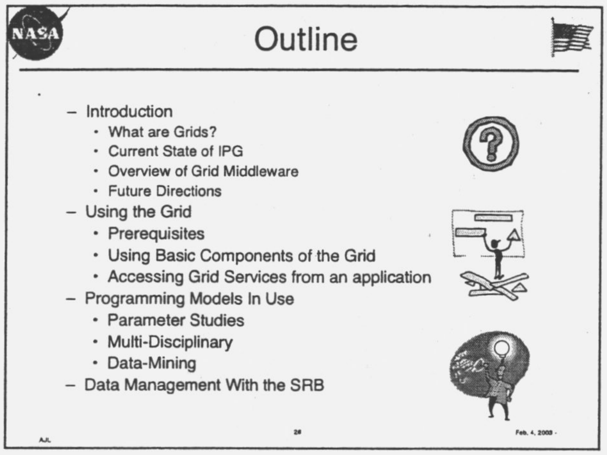
- Goal: Location-independent computing
- Intelligent resource selection
  - User-specified preferences and requirements
  - Resource characteristics, access, allocations, cost, load, performance predictions
- Create system-dependent job from system-independent job
  - Given a system (or systems) to use
  - Pick directories, set paths and environment, specify which executables and libraries to move, specify which data files to move
- Reliable execution of system-dependent job
  - Application execution & file management
- Dynamic access and accounting
  - Run applications without a permanent local account
  - Charging resource use to remote accounting systems
  - Resource pricing
- Workflow management
  - Specify and execute workflows
- Implemented as OGSA services



Data Grid

- Goal: Intelligently manage data sets in a grid
- Files identified by logical file names
- Files have physical locations
  - Called replicas
  - Location of replicas is maintained
  - User or tool can create or delete replicas
  - Replicas can be selected intelligently
  - Replicas can be managed intelligently
- Maintain collections of files
- Maintain metadata about files
- Access using libraries and command-line tools
  - May or may not resemble Unix file system usage
- Hear more during Storage Resource Broker (SRB) session of tutorial
- Important new area for the IPG

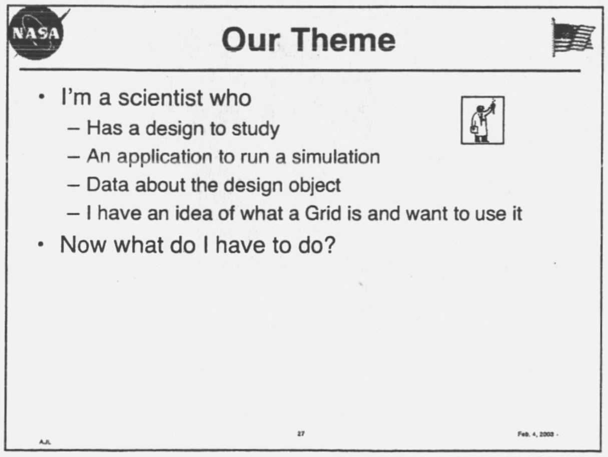
AJL 25 Feb. 4, 2003 -



Outline

-
  - Introduction
    - What are Grids?
    - Current State of IPG
    - Overview of Grid Middleware
    - Future Directions
  - Using the Grid
    - Prerequisites
    - Using Basic Components of the Grid
    - Accessing Grid Services from an application
  - Programming Models In Use
    - Parameter Studies
    - Multi-Disciplinary
    - Data-Mining
  - Data Management With the SRB

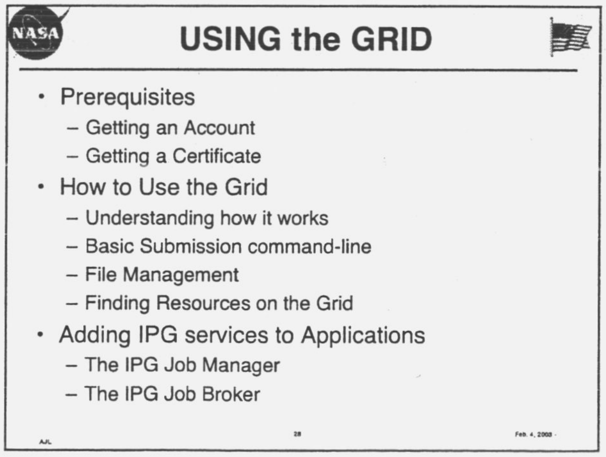
AJL 26 Feb. 4, 2003 -



Our Theme

- I'm a scientist who
  - Has a design to study
  - An application to run a simulation
  - Data about the design object
  - I have an idea of what a Grid is and want to use it
- Now what do I have to do?

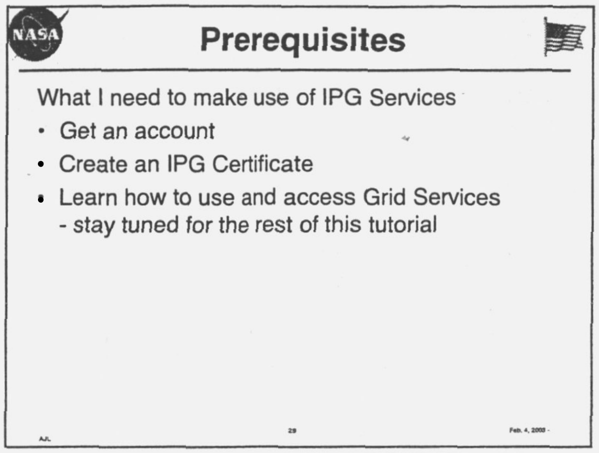
AJL 27 Feb. 4, 2003 -



USING the GRID

- Prerequisites
  - Getting an Account
  - Getting a Certificate
- How to Use the Grid
  - Understanding how it works
  - Basic Submission command-line
  - File Management
  - Finding Resources on the Grid
- Adding IPG services to Applications
  - The IPG Job Manager
  - The IPG Job Broker

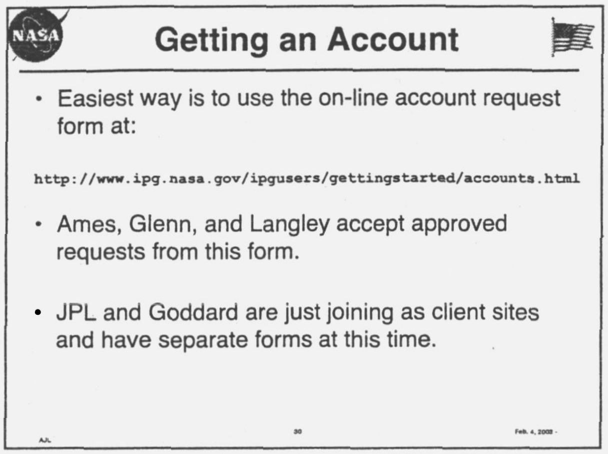
AJL 28 Feb. 4, 2003 -



Prerequisites

What I need to make use of IPG Services

- Get an account
- Create an IPG Certificate
- Learn how to use and access Grid Services
  - stay tuned for the rest of this tutorial



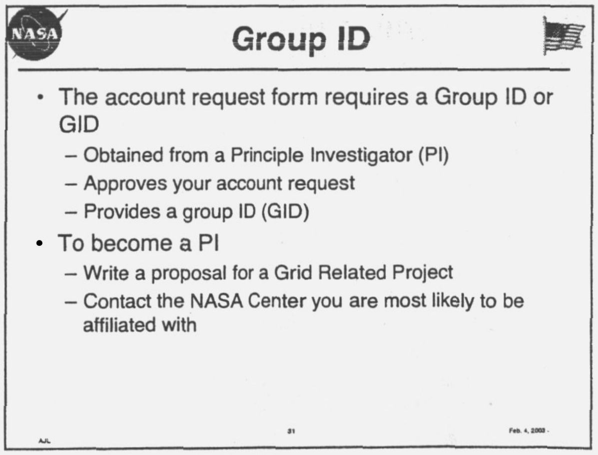
Getting an Account

- Easiest way is to use the on-line account request form at:

`http://www.ipg.nasa.gov/ipgusers/gettingstarted/accounts.html`

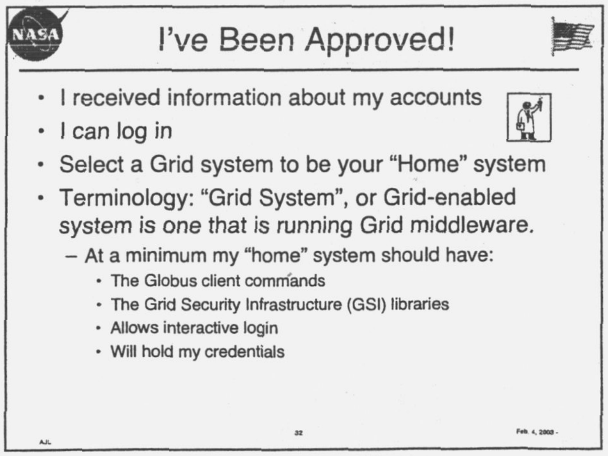
- Ames, Glenn, and Langley accept approved requests from this form.

- JPL and Goddard are just joining as client sites and have separate forms at this time.



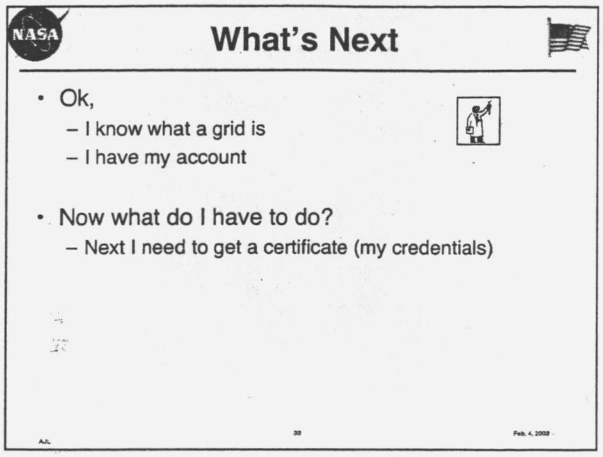
Group ID

- The account request form requires a Group ID or GID
  - Obtained from a Principle Investigator (PI)
  - Approves your account request
  - Provides a group ID (GID)
- To become a PI
  - Write a proposal for a Grid Related Project
  - Contact the NASA Center you are most likely to be affiliated with



I've Been Approved!

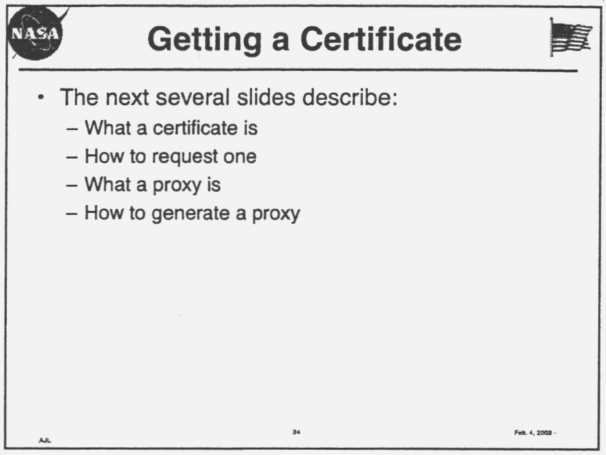
- I received information about my accounts
- I can log in
- Select a Grid system to be your "Home" system
- Terminology: "Grid System", or Grid-enabled system is one that is running Grid middleware.
  - At a minimum my "home" system should have:
    - The Globus client commands
    - The Grid Security Infrastructure (GSI) libraries
    - Allows interactive login
    - Will hold my credentials



What's Next

- Ok,
  - I know what a grid is
  - I have my account
- Now what do I have to do?
  - Next I need to get a certificate (my credentials)

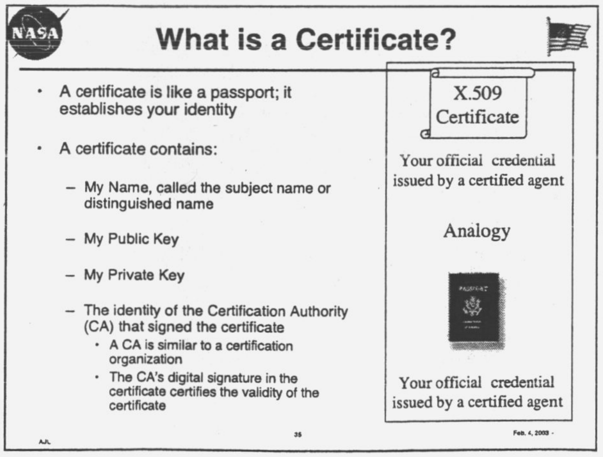
AJL 33 Feb. 4, 2003



Getting a Certificate

- The next several slides describe:
  - What a certificate is
  - How to request one
  - What a proxy is
  - How to generate a proxy

AJL 34 Feb. 4, 2003



What is a Certificate?

- A certificate is like a passport; it establishes your identity
- A certificate contains:
  - My Name, called the subject name or distinguished name
  - My Public Key
  - My Private Key
  - The identity of the Certification Authority (CA) that signed the certificate
    - A CA is similar to a certification organization
    - The CA's digital signature in the certificate certifies the validity of the certificate

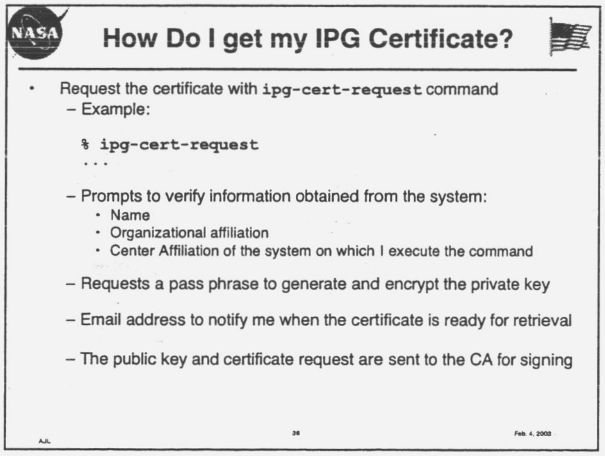
X.509 Certificate

Your official credential issued by a certified agent

Analogy

Your official credential issued by a certified agent

AJL 35 Feb. 4, 2003



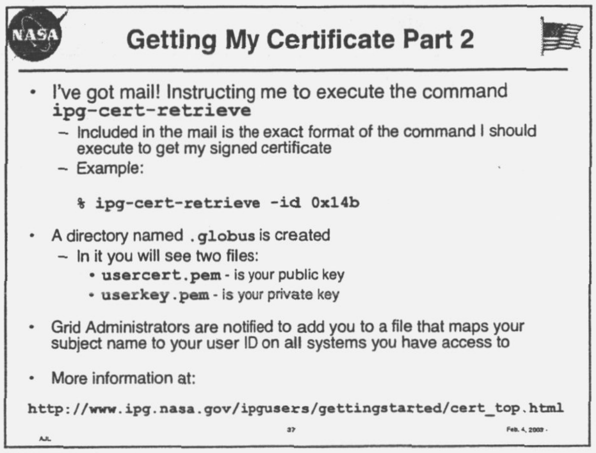
How Do I get my IPG Certificate?

- Request the certificate with `ipg-cert-request` command
  - Example:

```
% ipg-cert-request
...
```

  - Prompts to verify information obtained from the system:
    - Name
    - Organizational affiliation
    - Center Affiliation of the system on which I execute the command
  - Requests a pass phrase to generate and encrypt the private key
  - Email address to notify me when the certificate is ready for retrieval
  - The public key and certificate request are sent to the CA for signing

AJL 36 Feb. 4, 2003



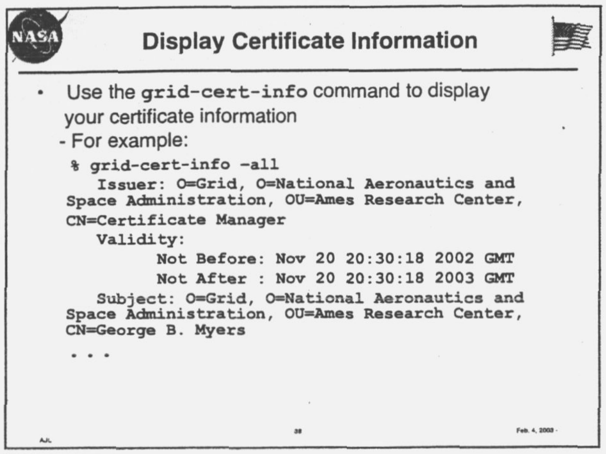
Getting My Certificate Part 2

- I've got mail! Instructing me to execute the command `ipg-cert-retrieve`
  - Included in the mail is the exact format of the command I should execute to get my signed certificate
  - Example:

```
% ipg-cert-retrieve -id 0x14b
```

- A directory named `.globus` is created
  - In it you will see two files:
    - `usercert.pem` - is your public key
    - `userkey.pem` - is your private key
- Grid Administrators are notified to add you to a file that maps your subject name to your user ID on all systems you have access to
- More information at:

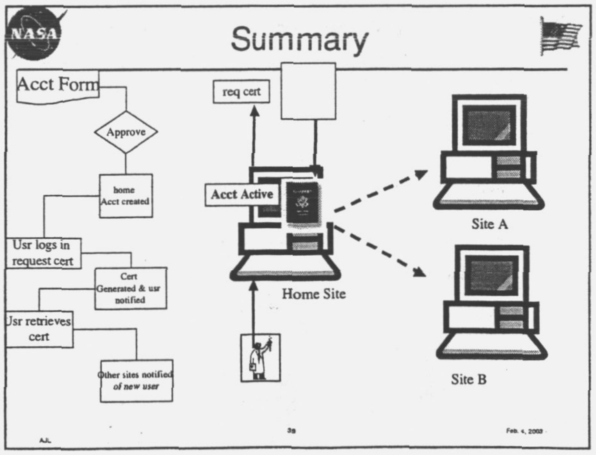
http://www.ipg.nasa.gov/ipgusers/gettingstarted/cert_top.html



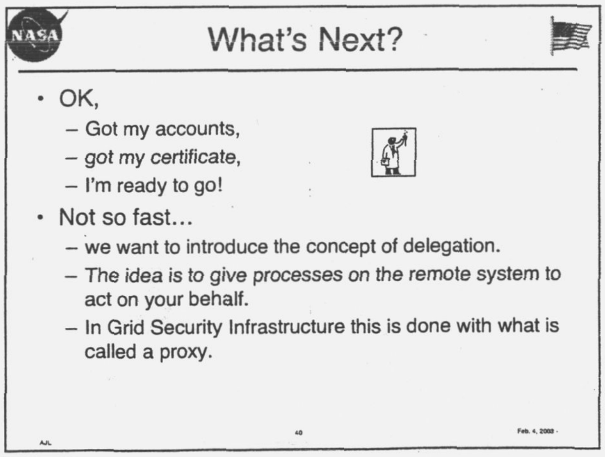
Display Certificate Information

- Use the `grid-cert-info` command to display your certificate information
  - For example:

```
% grid-cert-info -all
   Issuer: O=Grid, O=National Aeronautics and
Space Administration, OU=Ames Research Center,
CN=Certificate Manager
   Validity:
         Not Before: Nov 20 20:30:18 2002 GMT
         Not After : Nov 20 20:30:18 2003 GMT
   Subject: O=Grid, O=National Aeronautics and
Space Administration, OU=Ames Research Center,
CN=George B. Myers
. . .
```

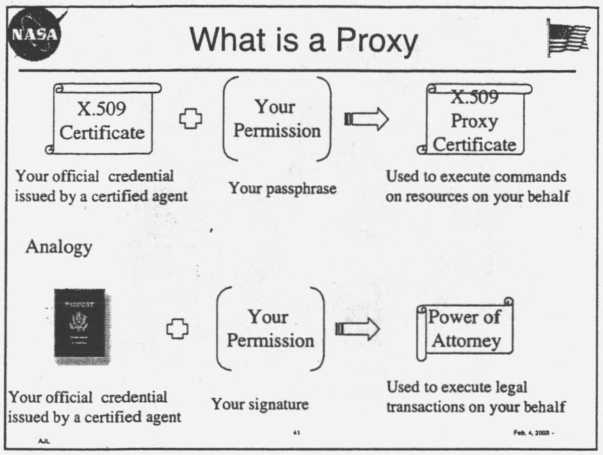


Summary



What's Next?

- OK,
  - Got my accounts,
  - *got my certificate,*
  - I'm ready to go!
- Not so fast…
  - we want to introduce the concept of delegation.
  - *The idea is to give processes on the remote system to* act on your behalf.
  - In Grid Security Infrastructure this is done with what is called a proxy.



What is a Proxy

Your official credential issued by a certified agent

Your passphrase

Used to execute commands on resources on your behalf

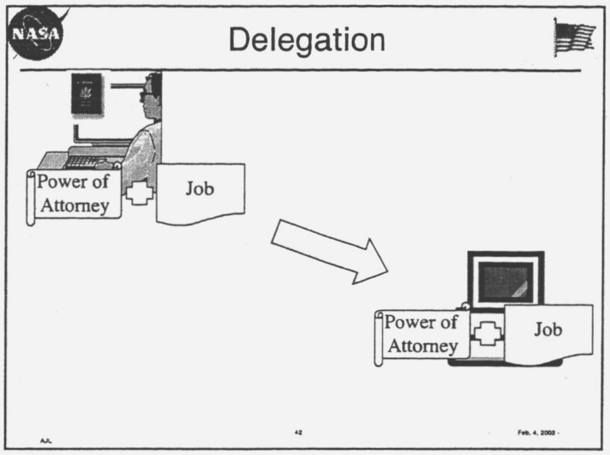
Analogy

Your official credential issued by a certified agent

Your signature

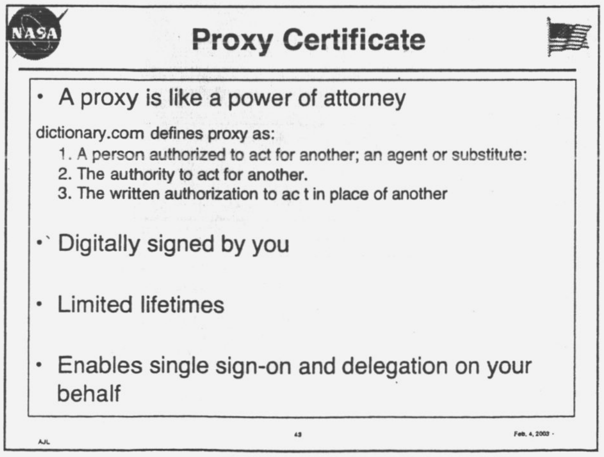
Used to execute legal transactions on your behalf

AJL 41 Feb. 4, 2003 -



Delegation

AJL 42 Feb. 4, 2003 -



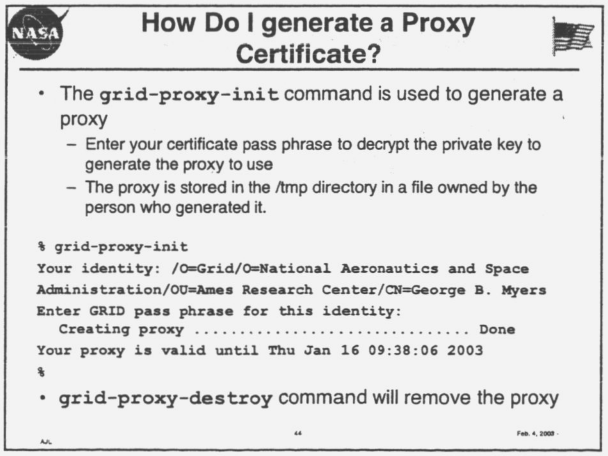
Proxy Certificate

- A proxy is like a power of attorney

dictionary.com defines proxy as:
1. A person authorized to act for another; an agent or substitute:
2. The authority to act for another.
3. The written authorization to ac t in place of another

- Digitally signed by you
- Limited lifetimes
- Enables single sign-on and delegation on your behalf

AJL 43 Feb. 4, 2003 -



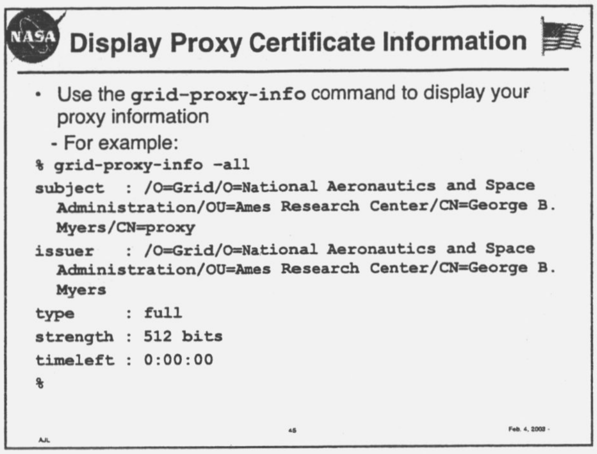
How Do I generate a Proxy Certificate?

- The `grid-proxy-init` command is used to generate a proxy
  - Enter your certificate pass phrase to decrypt the private key to generate the proxy to use
  - The proxy is stored in the /tmp directory in a file owned by the person who generated it.

```
% grid-proxy-init
Your identity: /O=Grid/O=National Aeronautics and Space
Administration/OU=Ames Research Center/CN=George B. Myers
Enter GRID pass phrase for this identity:
  Creating proxy .............................. Done
Your proxy is valid until Thu Jan 16 09:38:06 2003
%
```

- `grid-proxy-destroy` command will remove the proxy

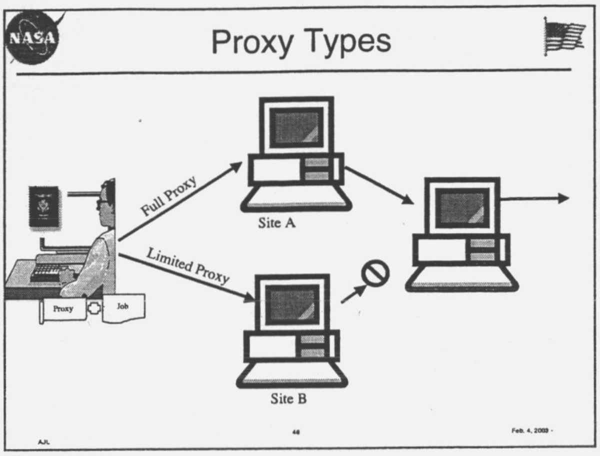
AJL 44 Feb. 4, 2003 -



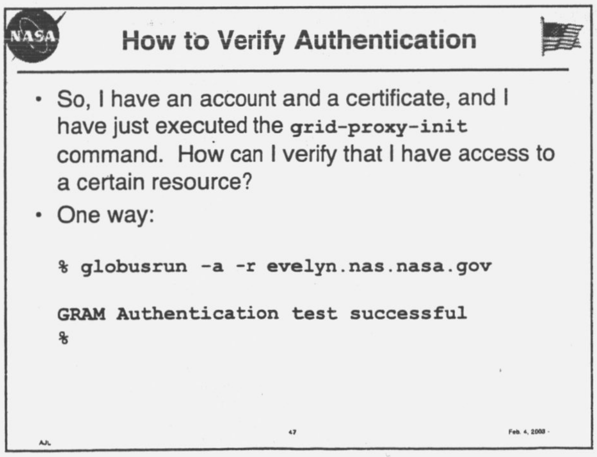
Display Proxy Certificate Information

- Use the `grid-proxy-info` command to display your proxy information
  - For example:

```
% grid-proxy-info -all
subject  : /O=Grid/O=National Aeronautics and Space
  Administration/OU=Ames Research Center/CN=George B.
  Myers/CN=proxy
issuer   : /O=Grid/O=National Aeronautics and Space
  Administration/OU=Ames Research Center/CN=George B.
  Myers
type     : full
strength : 512 bits
timeleft : 0:00:00
%
```



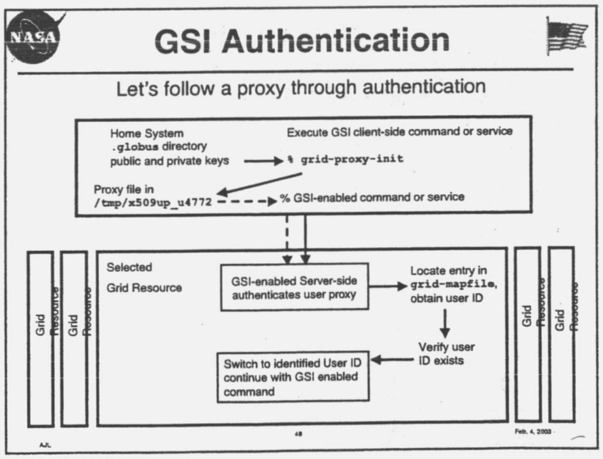
Proxy Types



How to Verify Authentication

- So, I have an account and a certificate, and I have just executed the `grid-proxy-init` command. How can I verify that I have access to a certain resource?
- One way:

```
% globusrun -a -r evelyn.nas.nasa.gov

GRAM Authentication test successful
%
```



GSI Authentication

Let's follow a proxy through authentication

Home System
`.globus` directory
public and private keys → `% grid-proxy-init`

Execute GSI client-side command or service

Proxy file in
`/tmp/x509up_u4772` → % GSI-enabled command or service

Selected
Grid Resource

GSI-enabled Server-side authenticates user proxy → Locate entry in `grid-mapfile`, obtain user ID → Verify user ID exists → Switch to identified User ID continue with GSI enabled command

While We're on Proxies

- Can I access the Grid from a Web Portal
  - The answer is yes, however a Portal needs a way to represent you
  - The MyProxy Service provides this capability

MyProxy

- A Client/Server method of storing a proxy for later retrieval from things like Web Portals
- Using the secure socket technology of the web a proxy is stored on a secured server
- The proxy has a limited life time and a different pass-phrase than that of your certificate.
- The proxy is retrieved by the Portal on your behalf and used in the same fashion as on any other Grid system.

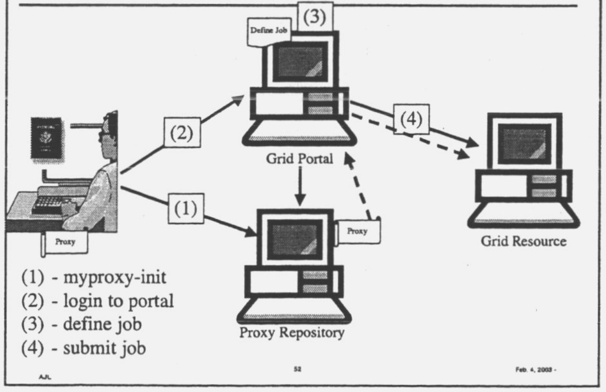
The `myproxy-init` command

- The **myproxy-init** command is executed on a Grid system where your certificate resides. Using SSL, and parameters and a passphrase you provide your proxy is stored on the MyProxy Server.
- Example:

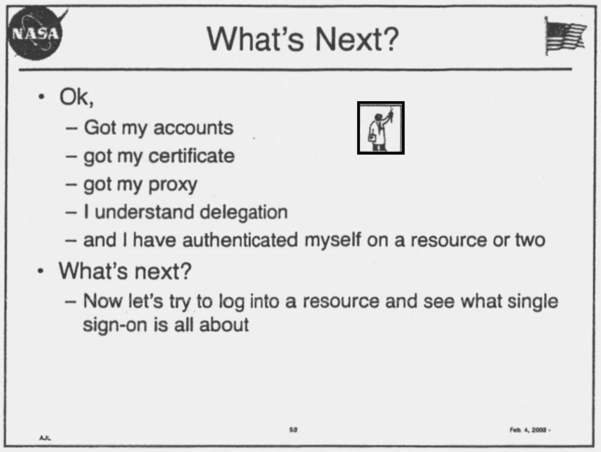
```
% myproxy-init
Your identity: /O=Grid/O=National Aeronautics and Space
   Administration/OU=Ames Research Center/CN=George B. Myers
Enter GRID pass phrase for this identity:
Creating proxy
   ................................................................ Done
Your proxy is valid until Thu Jan 23 16:40:36 2003
Enter MyProxy Pass Phrase:
Verifying password - Enter MyProxy Pass Phrase:
A proxy valid for 168 hours (  7.0 days  ) for user gmyers now exists on
   myproxy.ipg.nasa.gov.
%
```

- Also a **myproxy-destroy** command to remove your proxy from the Myproxy Server

MyProxy Overview



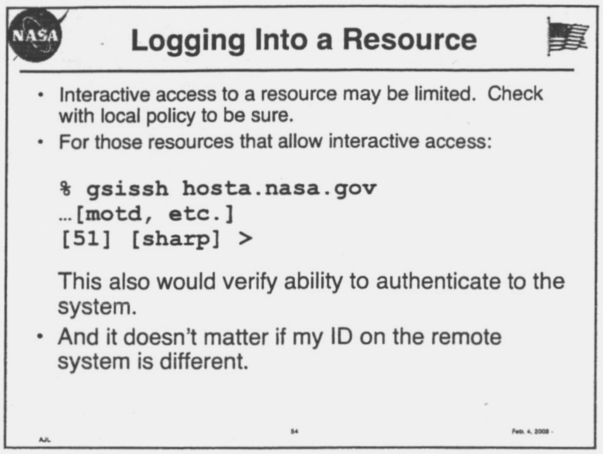
(1) - myproxy-init
(2) - login to portal
(3) - define job
(4) - submit job



What's Next?

- Ok,
  - Got my accounts
  - got my certificate
  - got my proxy
  - I understand delegation
  - and I have authenticated myself on a resource or two
- What's next?
  - Now let's try to log into a resource and see what single sign-on is all about

AJL 53 Feb. 4, 2003 -



Logging Into a Resource

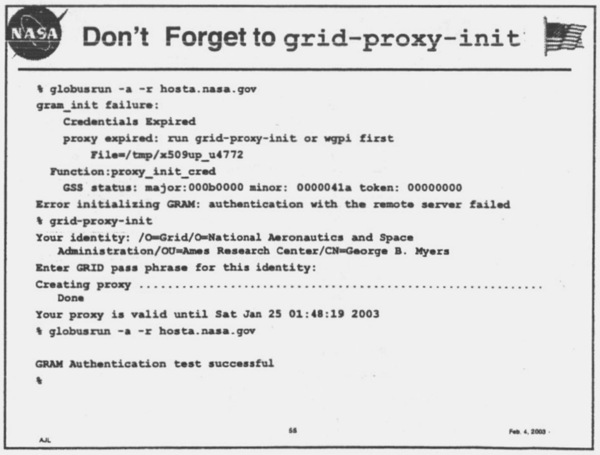
- Interactive access to a resource may be limited. Check with local policy to be sure.
- For those resources that allow interactive access:

```
% gsissh hosta.nasa.gov
…[motd, etc.]
[51] [sharp] >
```

This also would verify ability to authenticate to the system.

- And it doesn't matter if my ID on the remote system is different.

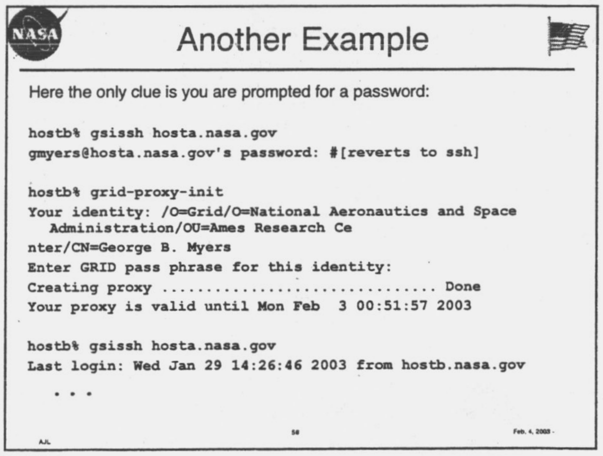
AJL 54 Feb. 4, 2003 -



Don't Forget to `grid-proxy-init`

```
% globusrun -a -r hosta.nasa.gov
gram_init failure:
    Credentials Expired
    proxy expired: run grid-proxy-init or wgpi first
        File=/tmp/x509up_u4772
  Function:proxy_init_cred
    GSS status: major:000b0000 minor: 0000041a token: 00000000
Error initializing GRAM: authentication with the remote server failed
% grid-proxy-init
Your identity: /O=Grid/O=National Aeronautics and Space
   Administration/OU=Ames Research Center/CN=George B. Myers
Enter GRID pass phrase for this identity:
Creating proxy ............................................................
   Done
Your proxy is valid until Sat Jan 25 01:48:19 2003
% globusrun -a -r hosta.nasa.gov

GRAM Authentication test successful
%
```

AJL 55 Feb. 4, 2003 -

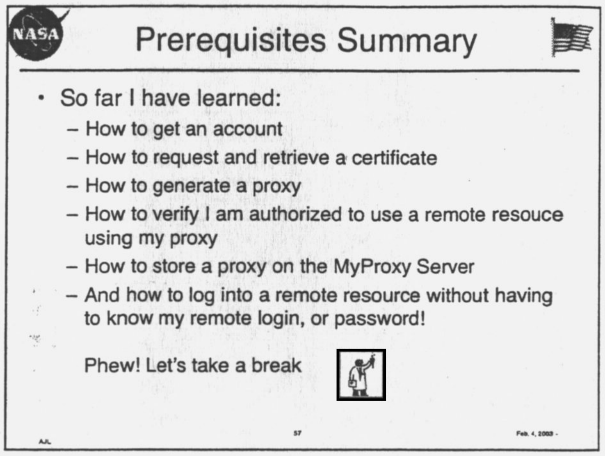


Another Example

Here the only clue is you are prompted for a password:

```
hostb% gsissh hosta.nasa.gov
gmyers@hosta.nasa.gov's password: #[reverts to ssh]

hostb% grid-proxy-init
Your identity: /O=Grid/O=National Aeronautics and Space
  Administration/OU=Ames Research Ce
nter/CN=George B. Myers
Enter GRID pass phrase for this identity:
Creating proxy .............................. Done
Your proxy is valid until Mon Feb  3 00:51:57 2003

hostb% gsissh hosta.nasa.gov
Last login: Wed Jan 29 14:26:46 2003 from hostb.nasa.gov
   . . .
```

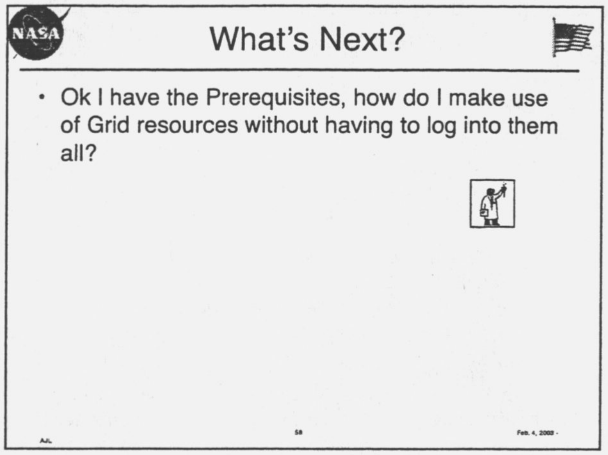
AJL 56 Feb. 4, 2003 -



Prerequisites Summary

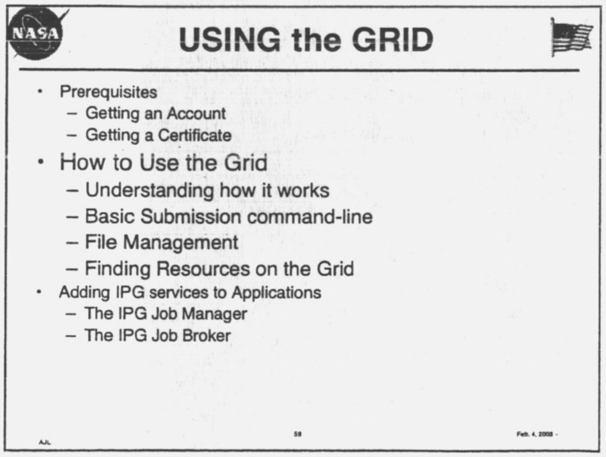
- So far I have learned:
  - How to get an account
  - How to request and retrieve a certificate
  - How to generate a proxy
  - How to verify I am authorized to use a remote resouce using my proxy
  - How to store a proxy on the MyProxy Server
  - And how to log into a remote resource without having to know my remote login, or password!

  Phew! Let's take a break



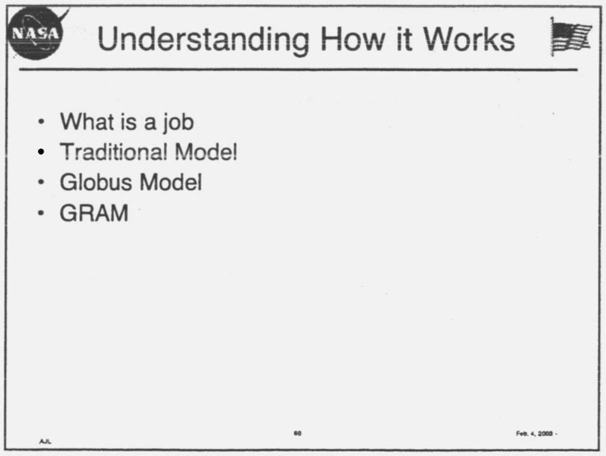
What's Next?

- Ok I have the Prerequisites, how do I make use of Grid resources without having to log into them all?



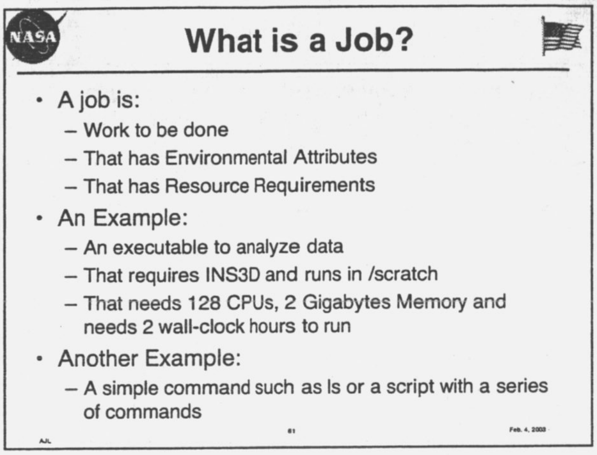
USING the GRID

- Prerequisites
  - Getting an Account
  - Getting a Certificate
- How to Use the Grid
  - Understanding how it works
  - Basic Submission command-line
  - File Management
  - Finding Resources on the Grid
- Adding IPG services to Applications
  - The IPG Job Manager
  - The IPG Job Broker



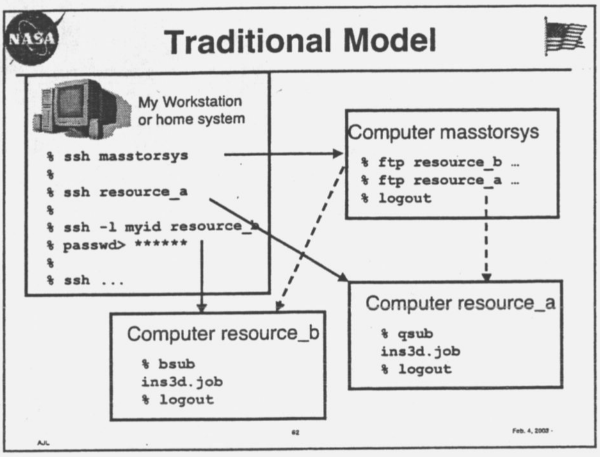
Understanding How it Works

- What is a job
- Traditional Model
- Globus Model
- GRAM



What is a Job?

- A job is:
  - Work to be done
  - That has Environmental Attributes
  - That has Resource Requirements
- An Example:
  - An executable to analyze data
  - That requires INS3D and runs in /scratch
  - That needs 128 CPUs, 2 Gigabytes Memory and needs 2 wall-clock hours to run
- Another Example:
  - A simple command such as ls or a script with a series of commands



Traditional Model

My Workstation or home system

```
% ssh masstorsys
%
% ssh resource_a
%
% ssh -l myid resource_b
% passwd> ******
%
% ssh ...
```

Computer masstorsys

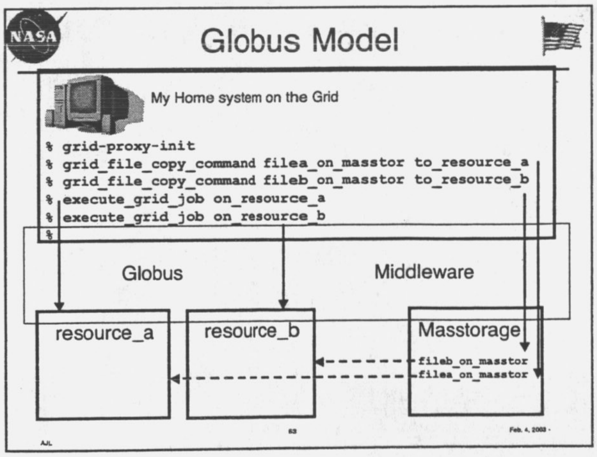
```
% ftp resource_b ...
% ftp resource_a ...
% logout
```

Computer resource_a

```
% qsub
ins3d.job
% logout
```

Computer resource_b

```
% bsub
ins3d.job
% logout
```



Globus Model

My Home system on the Grid

```
% grid-proxy-init
% grid_file_copy_command filea_on_masstor to_resource_a
% grid_file_copy_command fileb_on_masstor to_resource_b
% execute_grid_job on_resource_a
% execute_grid_job on_resource_b
%
```

Globus Middleware

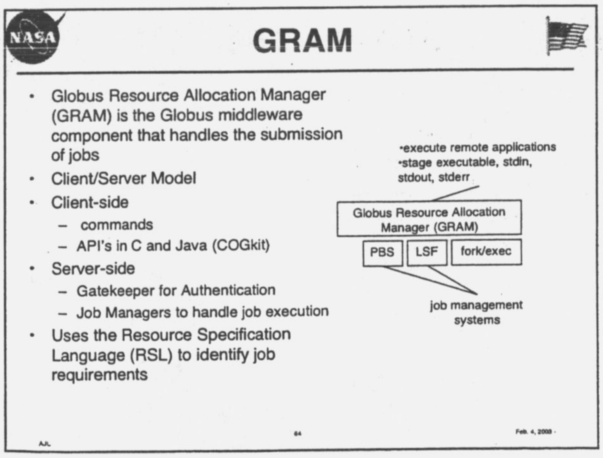
resource_a

resource_b

Masstorage

fileb_on_masstor

filea_on_masstor



GRAM

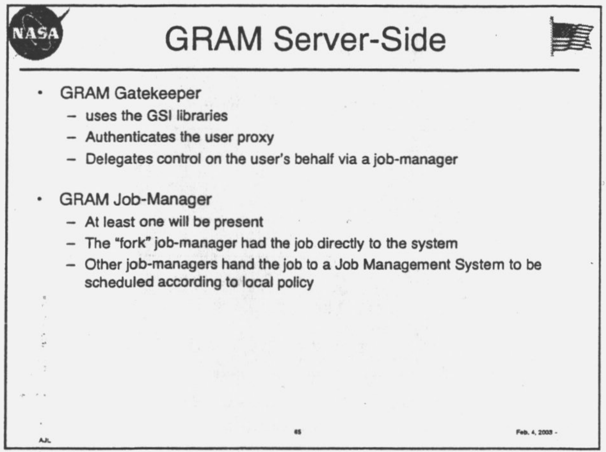
- Globus Resource Allocation Manager (GRAM) is the Globus middleware component that handles the submission of jobs
- Client/Server Model
- Client-side
  - commands
  - API's in C and Java (COGkit)
- Server-side
  - Gatekeeper for Authentication
  - Job Managers to handle job execution
- Uses the Resource Specification Language (RSL) to identify job requirements

•execute remote applications
•stage executable, stdin, stdout, stderr

Globus Resource Allocation Manager (GRAM)

PBS | LSF | fork/exec

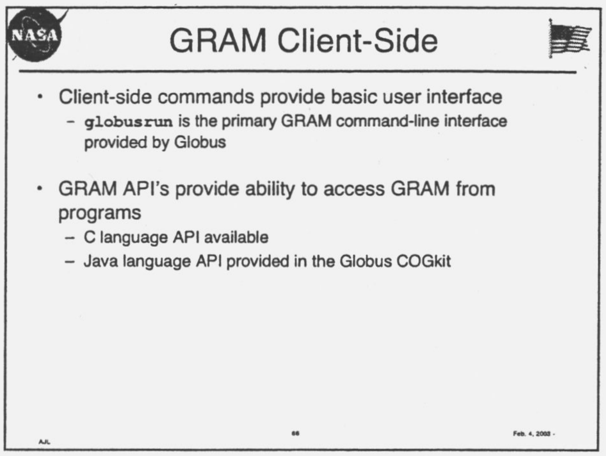
job management systems



GRAM Server-Side

- GRAM Gatekeeper
  - uses the GSI libraries
  - Authenticates the user proxy
  - Delegates control on the user's behalf via a job-manager
- GRAM Job-Manager
  - At least one will be present
  - The "fork" job-manager had the job directly to the system
  - Other job-managers hand the job to a Job Management System to be scheduled according to local policy

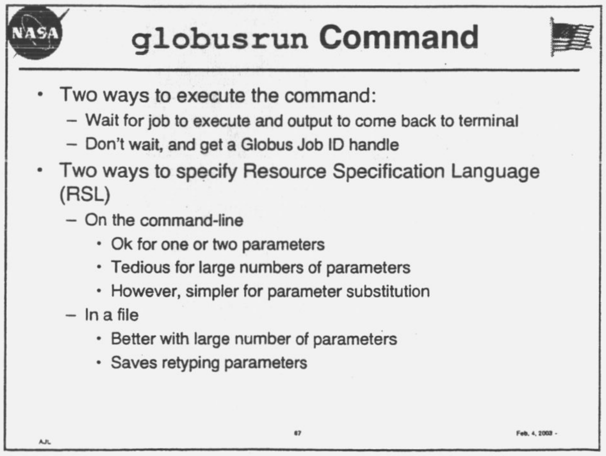
AJL 65 Feb. 4, 2003 -



GRAM Client-Side

- Client-side commands provide basic user interface
  - `globusrun` is the primary GRAM command-line interface provided by Globus
- GRAM API's provide ability to access GRAM from programs
  - C language API available
  - Java language API provided in the Globus COGkit

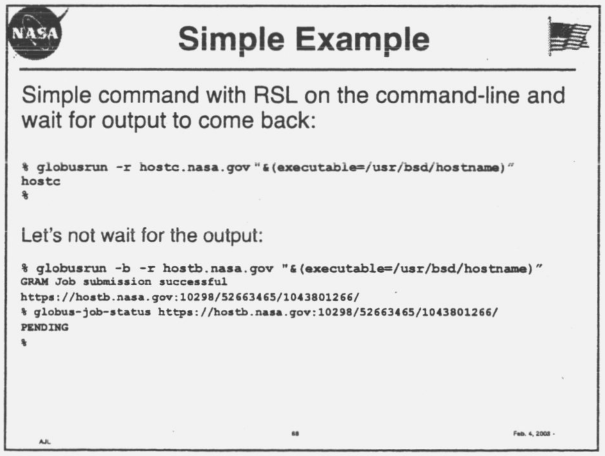
AJL 66 Feb. 4, 2003 -



`globusrun` Command

- Two ways to execute the command:
  - Wait for job to execute and output to come back to terminal
  - Don't wait, and get a Globus Job ID handle
- Two ways to specify Resource Specification Language (RSL)
  - On the command-line
    - Ok for one or two parameters
    - Tedious for large numbers of parameters
    - However, simpler for parameter substitution
  - In a file
    - Better with large number of parameters
    - Saves retyping parameters

AJL 67 Feb. 4, 2003 -



Simple Example

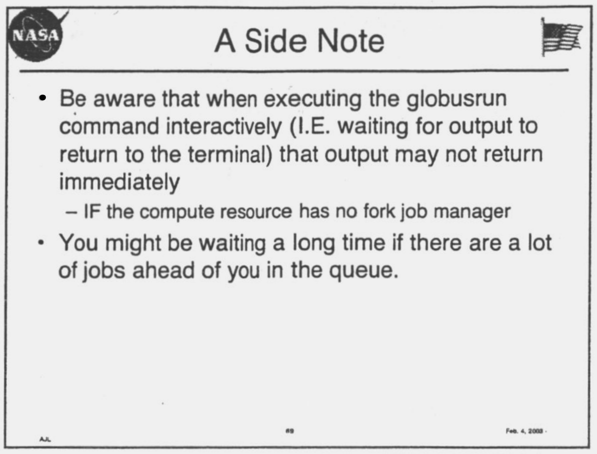
Simple command with RSL on the command-line and wait for output to come back:

```
% globusrun -r hostc.nasa.gov "&(executable=/usr/bsd/hostname)"
hostc
%
```

Let's not wait for the output:

```
% globusrun -b -r hostb.nasa.gov "&(executable=/usr/bsd/hostname)"
GRAM Job submission successful
https://hostb.nasa.gov:10298/52663465/1043801266/
% globus-job-status https://hostb.nasa.gov:10298/52663465/1043801266/
PENDING
%
```

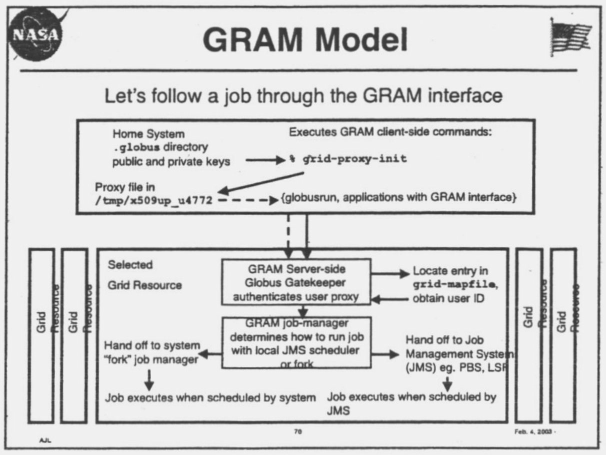
AJL 68 Feb. 4, 2003 -



A Side Note

- Be aware that when executing the globusrun command interactively (I.E. waiting for output to return to the terminal) that output may not return immediately
  - IF the compute resource has no fork job manager
- You might be waiting a long time if there are a lot of jobs ahead of you in the queue.

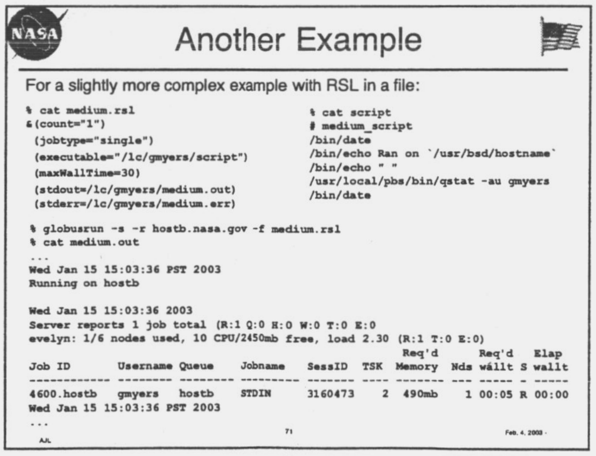
AJL 69 Feb. 4, 2003



GRAM Model

Let's follow a job through the GRAM interface

AJL 70 Feb. 4, 2003

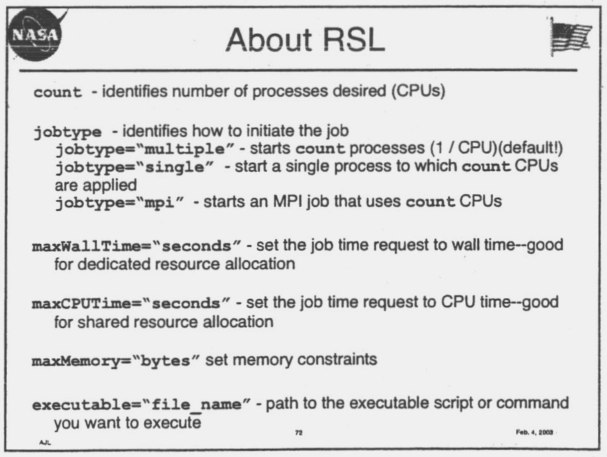


Another Example

For a slightly more complex example with RSL in a file:

```
% cat medium.rsl                           % cat script
&(count="1")                               # medium_script
 (jobtype="single")                        /bin/date
 (executable="/lc/gmyers/script")          /bin/echo Ran on `/usr/bsd/hostname`
 (maxWallTime=30)                          /bin/echo " "
 (stdout=/lc/gmyers/medium.out)            /usr/local/pbs/bin/qstat -au gmyers
 (stderr=/lc/gmyers/medium.err)            /bin/date

% globusrun -s -r hostb.nasa.gov -f medium.rsl
% cat medium.out
...
Wed Jan 15 15:03:36 PST 2003
Running on hostb

Wed Jan 15 15:03:36 2003
Server reports 1 job total (R:1 Q:0 H:0 W:0 T:0 E:0
evelyn: 1/6 nodes used, 10 CPU/2450mb free, load 2.30 (R:1 T:0 E:0)
                                                         Req'd       Req'd   Elap
Job ID       Username Queue    Jobname    SessID  TSK  Memory  Nds wállt S wallt
------------ -------- -------- ---------- ------- ---- ------- --- ----- - -----
4600.hostb   gmyers   hostb    STDIN      3160473    2  490mb    1 00:05 R 00:00
Wed Jan 15 15:03:36 PST 2003
...
```

AJL 71 Feb. 4, 2003



About RSL

`count` - identifies number of processes desired (CPUs)

`jobtype` - identifies how to initiate the job
`jobtype="multiple"` - starts `count` processes (1 / CPU)(default!)
`jobtype="single"` - start a single process to which `count` CPUs are applied
`jobtype="mpi"` - starts an MPI job that uses `count` CPUs

`maxWallTime="seconds"` - set the job time request to wall time--good for dedicated resource allocation

`maxCPUTime="seconds"` - set the job time request to CPU time--good for shared resource allocation

`maxMemory="bytes"` set memory constraints

`executable="file_name"` - path to the executable script or command you want to execute

AJL 72 Feb. 4, 2003

*Comments on RSL

- An **executable** can be anything from a command, application or script
- **Jobtype=multiple** might be used where you have your own inter-process communication
- **jobtype=single** would be used when using OpenMP, and other multithreading technologies
- Be aware that the use of **maxMemory** is not consistent through the Grid. Some sites consider it to mean memory per job, while others consider it to mean per process.

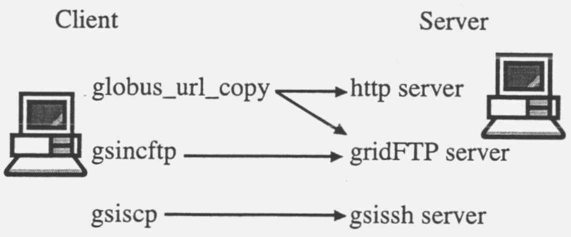
What's Next

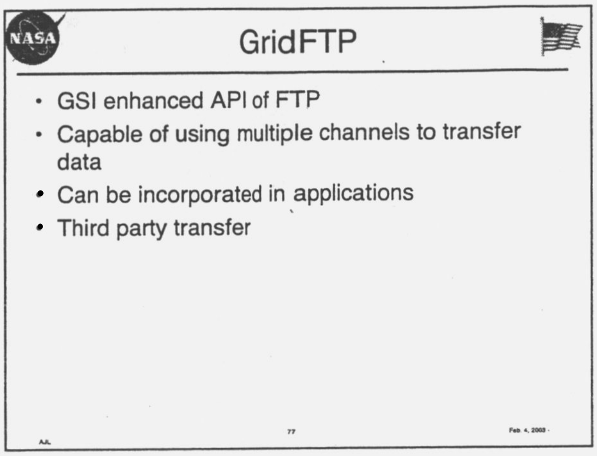
- Ok,
  - I now know how to submit a job to a grid resource
  - I know more than I ever wanted about GRAM
  - I know what to expect when I execute globusrun
  - I know about RSL and how to use it
- Now What?
  - How do I move my data around in the environment?
    - Its simple,
      - There exists modified FTP and SCP commands
      - Plus a client that allows 3rd party transfer

File Transfer Methods

- Grid-enabled File transfer protocol API - GridFTP
- gsincftp - NCFTP with GSI that uses GridFTP
- globus-url-copy - enabling third party transfers
- gsiscp - GSI enabled SSH copy command

Data Movement

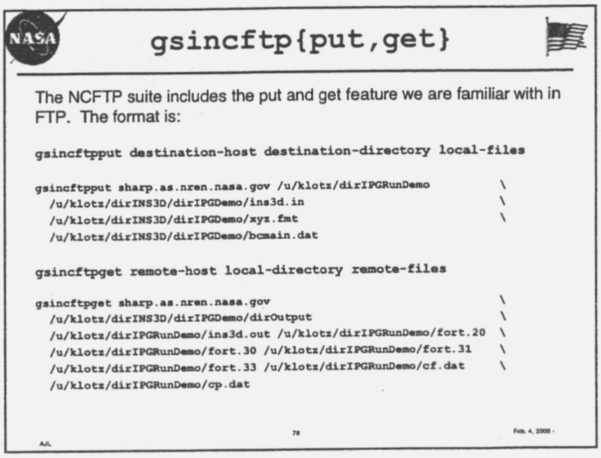




GridFTP

- GSI enhanced API of FTP
- Capable of using multiple channels to transfer data
- Can be incorporated in applications
- Third party transfer

AJL 77 Feb. 4, 2003 -

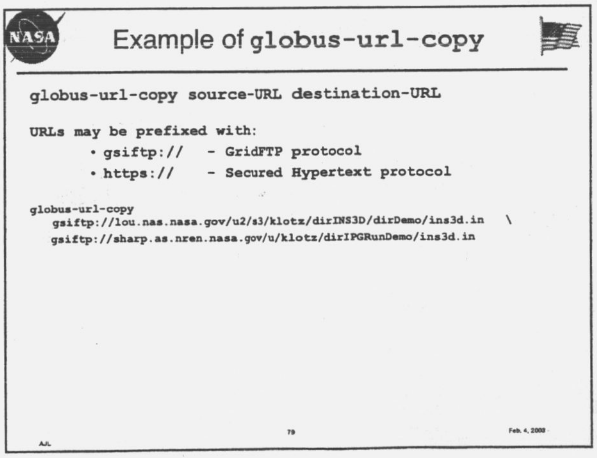


gsincftp{put,get}

The NCFTP suite includes the put and get feature we are familiar with in FTP. The format is:

```
gsincftpput destination-host destination-directory local-files

gsincftpput sharp.as.nren.nasa.gov /u/klotz/dirIPGRunDemo          \
  /u/klotz/dirINS3D/dirIPGDemo/ins3d.in                            \
  /u/klotz/dirINS3D/dirIPGDemo/xyz.fmt                             \
  /u/klotz/dirINS3D/dirIPGDemo/bcmain.dat

gsincftpget remote-host local-directory remote-files

gsincftpget sharp.as.nren.nasa.gov                                 \
  /u/klotz/dirINS3D/dirIPGDemo/dirOutput                           \
  /u/klotz/dirIPGRunDemo/ins3d.out /u/klotz/dirIPGRunDemo/fort.20  \
  /u/klotz/dirIPGRunDemo/fort.30 /u/klotz/dirIPGRunDemo/fort.31    \
  /u/klotz/dirIPGRunDemo/fort.33 /u/klotz/dirIPGRunDemo/cf.dat     \
  /u/klotz/dirIPGRunDemo/cp.dat
```

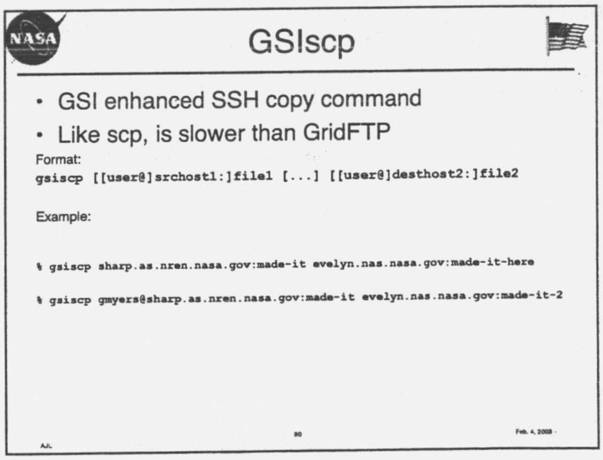
AJL 78 Feb. 4, 2003 -



Example of globus-url-copy

```
globus-url-copy source-URL destination-URL

URLs may be prefixed with:
        • gsiftp://   - GridFTP protocol
        • https://    - Secured Hypertext protocol

globus-url-copy
   gsiftp://lou.nas.nasa.gov/u2/s3/klotz/dirINS3D/dirDemo/ins3d.in   \
   gsiftp://sharp.as.nren.nasa.gov/u/klotz/dirIPGRunDemo/ins3d.in
```

AJL 79 Feb. 4, 2003



GSIscp

- GSI enhanced SSH copy command
- Like scp, is slower than GridFTP

Format:

```
gsiscp [[user@]srchost1:]file1 [...] [[user@]desthost2:]file2
```

Example:

```
% gsiscp sharp.as.nren.nasa.gov:made-it evelyn.nas.nasa.gov:made-it-here

% gsiscp gmyers@sharp.as.nren.nasa.gov:made-it evelyn.nas.nasa.gov:made-it-2
```

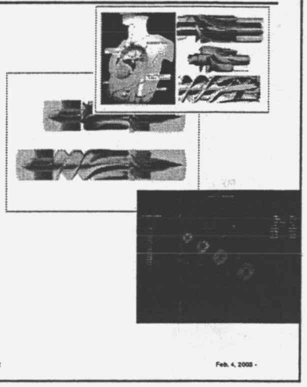
AJL 80 Feb. 4, 2003 -

What's Next

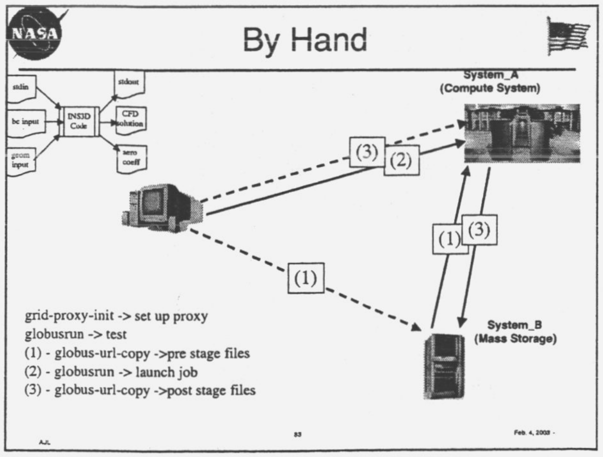
- OK,
  - I understand how the basics work
  - I know how to submit a job
  - I know how to move my data
- What's Next?
  - Putting it all together.
    - Creating scripts to allow me to stage my actual input files, run my application, and get my results.

INS3D

- has been used in numerous applications,
  - analysis of the fuel turbopump for the Space Shuttle's main engines.
  - to make major design improvements to a heart device that could save thousands of lives each year. tDeBakey Ventricular Assist Device
- INS3D code solves the incompressible Navier-Stokes equations in three-dimensional generalized coordinates for both steady-state and time varying flow.
- It is written in Fortran 77, with some extensions and several routines in C.
- The source code comes ready to compile and run on either a Cray computer or a Silicon Graphics workstation.



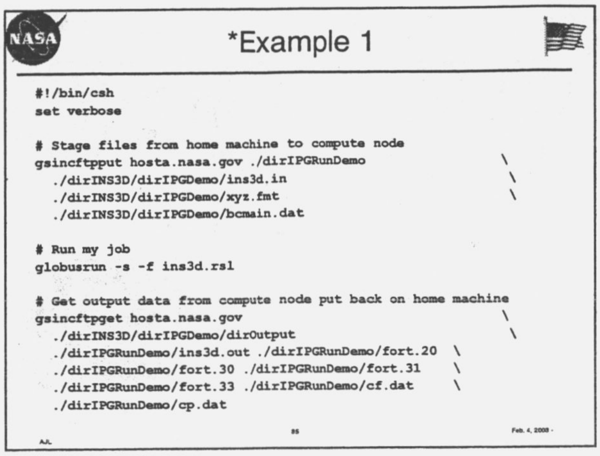
By Hand



grid-proxy-init -> set up proxy
globusrun -> test
(1) - globus-url-copy ->pre stage files
(2) - globusrun -> launch job
(3) - globus-url-copy ->post stage files

Basic RSL for INS3D

```
% cat ins3d.rsl
+
( &(resourceManagerContact="hosta.nasa.gov")
  (count=1)
  (maxWallTime=10)
  (directory=./dirIPGRunDemo)
  (executable=$(GLOBUSRUN_GASS_URL)./dirINS3D/dirIPGDemo/ins3d_dp)
  (stdin=ins3d.in)
  (stderr=ins3d.err)
  (stdout=ins3d.out)
)
```

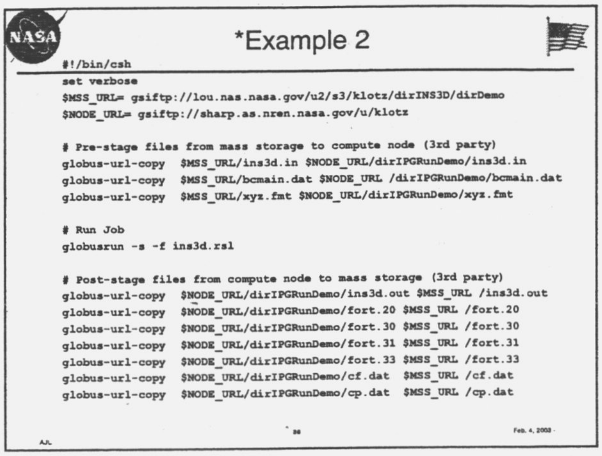


*Example 1

```
#!/bin/csh
set verbose

# Stage files from home machine to compute node
gsincftpput hosta.nasa.gov ./dirIPGRunDemo                      \
   ./dirINS3D/dirIPGDemo/ins3d.in                                \
   ./dirINS3D/dirIPGDemo/xyz.fmt                                 \
   ./dirINS3D/dirIPGDemo/bcmain.dat

# Run my job
globusrun -s -f ins3d.rsl

# Get output data from compute node put back on home machine
gsincftpget hosta.nasa.gov                                      \
   ./dirINS3D/dirIPGDemo/dirOutput                               \
   ./dirIPGRunDemo/ins3d.out ./dirIPGRunDemo/fort.20  \
   ./dirIPGRunDemo/fort.30 ./dirIPGRunDemo/fort.31    \
   ./dirIPGRunDemo/fort.33 ./dirIPGRunDemo/cf.dat     \
   ./dirIPGRunDemo/cp.dat
```

85 Feb. 4, 2003 -

AJL

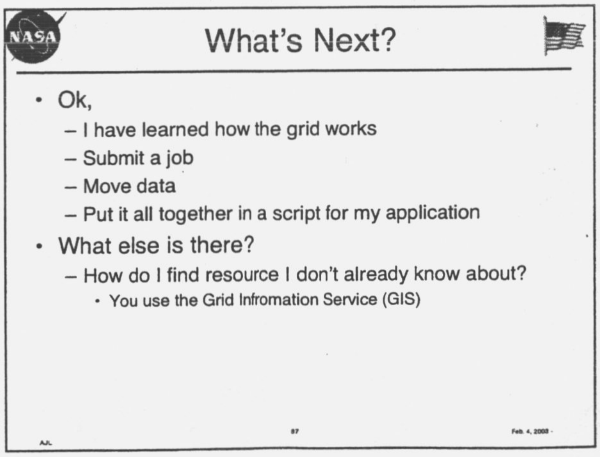


*Example 2

```
#!/bin/csh
set verbose
$MSS_URL= gsiftp://lou.nas.nasa.gov/u2/s3/klotz/dirINS3D/dirDemo
$NODE_URL= gsiftp://sharp.as.nren.nasa.gov/u/klotz

# Pre-stage files from mass storage to compute node (3rd party)
globus-url-copy  $MSS_URL/ins3d.in $NODE_URL/dirIPGRunDemo/ins3d.in
globus-url-copy  $MSS_URL/bcmain.dat $NODE_URL /dirIPGRunDemo/bcmain.dat
globus-url-copy  $MSS_URL/xyz.fmt $NODE_URL/dirIPGRunDemo/xyz.fmt

# Run Job
globusrun -s -f ins3d.rsl

# Post-stage files from compute node to mass storage (3rd party)
globus-url-copy  $NODE_URL/dirIPGRunDemo/ins3d.out $MSS_URL /ins3d.out
globus-url-copy  $NODE_URL/dirIPGRunDemo/fort.20 $MSS_URL /fort.20
globus-url-copy  $NODE_URL/dirIPGRunDemo/fort.30 $MSS_URL /fort.30
globus-url-copy  $NODE_URL/dirIPGRunDemo/fort.31 $MSS_URL /fort.31
globus-url-copy  $NODE_URL/dirIPGRunDemo/fort.33 $MSS_URL /fort.33
globus-url-copy  $NODE_URL/dirIPGRunDemo/cf.dat  $MSS_URL /cf.dat
globus-url-copy  $NODE_URL/dirIPGRunDemo/cp.dat  $MSS_URL /cp.dat
```

86 Feb. 4, 2003 -

AJL

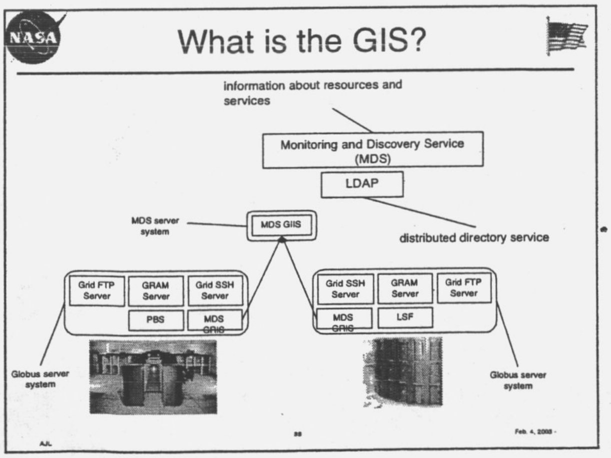


What's Next?

- Ok,
  - – I have learned how the grid works
  - – Submit a job
  - – Move data
  - – Put it all together in a script for my application
- What else is there?
  - – How do I find resource I don't already know about?
    - You use the Grid Infromation Service (GIS)

87 Feb. 4, 2003 -

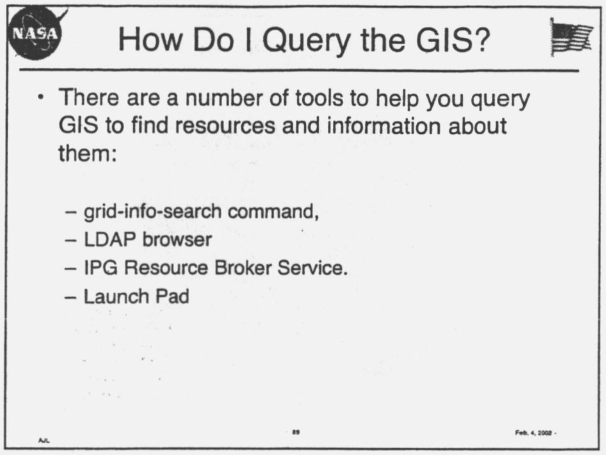
AJL



What is the GIS?

88 Feb. 4, 2003 -

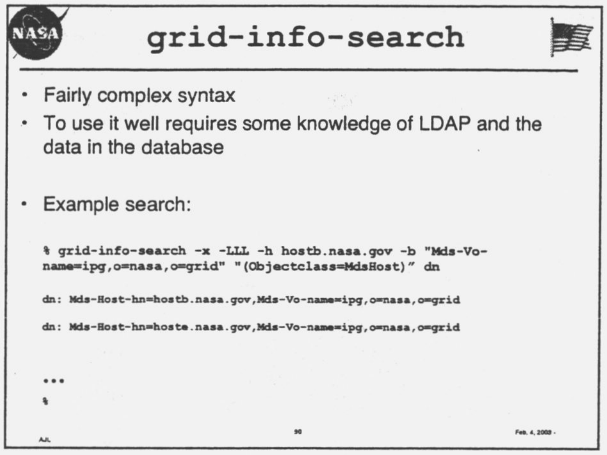
AJL



How Do I Query the GIS?

- There are a number of tools to help you query GIS to find resources and information about them:
  - grid-info-search command,
  - LDAP browser
  - IPG Resource Broker Service.
  - Launch Pad

AJL 89 Feb. 4, 2003 -

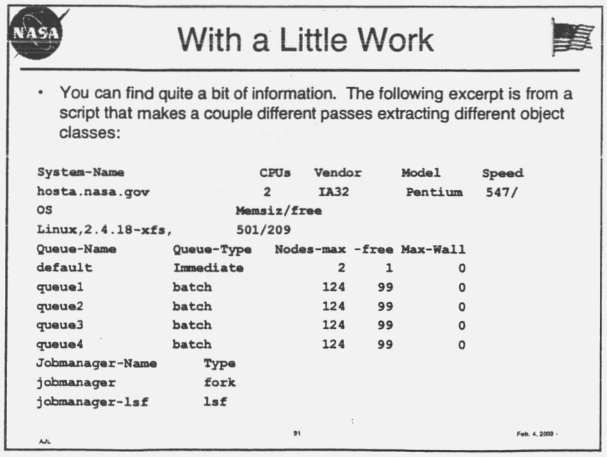


grid-info-search

- Fairly complex syntax
- To use it well requires some knowledge of LDAP and the data in the database
- Example search:

```
% grid-info-search -x -LLL -h hostb.nasa.gov -b "Mds-Vo-
name=ipg,o=nasa,o=grid" "(Objectclass=MdsHost)" dn

dn: Mds-Host-hn=hostb.nasa.gov,Mds-Vo-name=ipg,o=nasa,o=grid

dn: Mds-Host-hn=hoste.nasa.gov,Mds-Vo-name=ipg,o=nasa,o=grid


...
%
```

AJL 90 Feb. 4, 2003 -

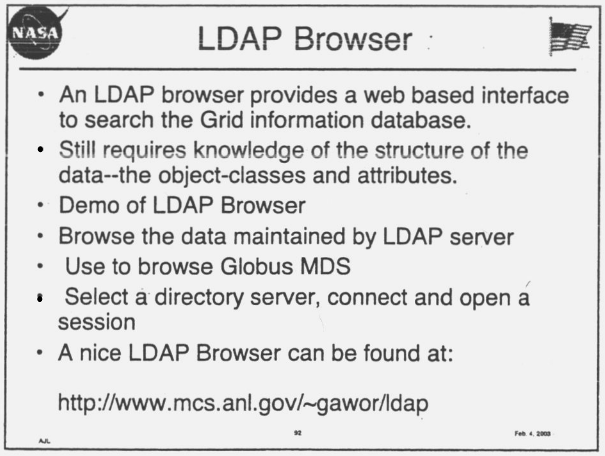


With a Little Work

- You can find quite a bit of information. The following excerpt is from a script that makes a couple different passes extracting different object classes:

```
System-Name                    CPUs  Vendor     Model      Speed
hosta.nasa.gov                  2     IA32       Pentium   547/
OS                           Memsiz/free
Linux,2.4.18-xfs,            501/209
Queue-Name        Queue-Type   Nodes-max -free Max-Wall
default           Immediate           2     1        0
queue1            batch             124    99        0
queue2            batch             124    99        0
queue3            batch             124    99        0
queue4            batch             124    99        0
Jobmanager-Name      Type
jobmanager           fork
jobmanager-lsf       lsf
```

AJL 91 Feb. 4, 2003 -



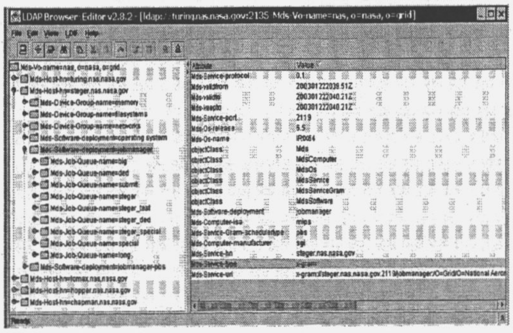
LDAP Browser

- An LDAP browser provides a web based interface to search the Grid information database.
- Still requires knowledge of the structure of the data--the object-classes and attributes.
- Demo of LDAP Browser
- Browse the data maintained by LDAP server
- Use to browse Globus MDS
- Select a directory server, connect and open a session
- A nice LDAP Browser can be found at:

  http://www.mcs.anl.gov/~gawor/ldap

AJL 92 Feb. 4, 2003

LDAP Browser – MDS Data

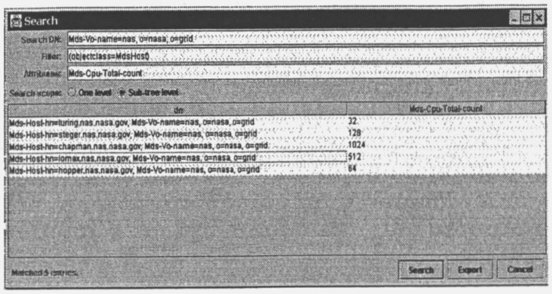


LDAP Browser Editor v2.8.2 - [ldap://turing.nas.nasa.gov:2135 Mds-Vo-name=nas, o=nasa, o=grid]

```
Mds-Vo-name=nas, o=nasa, o=grid
  Mds-Host-hn=turing.nas.nasa.gov
  Mds-Host-hn=steger.nas.nasa.gov
    Mds-Device-Group-name=memory
    Mds-Device-Group-name=filesystems
    Mds-Device-Group-name=networks
    Mds-Software-deployment=operating system
    Mds-Software-deployment=jobmanager
      Mds-Job-Queue-name=big
      Mds-Job-Queue-name=ded
      Mds-Job-Queue-name=submit
      Mds-Job-Queue-name=steger
      Mds-Job-Queue-name=steger_test
      Mds-Job-Queue-name=steger_ded
      Mds-Job-Queue-name=steger_special
      Mds-Job-Queue-name=special
      Mds-Job-Queue-name=long
    Mds-Software-deployment=jobmanager-pbs
  Mds-Host-hn=lomax.nas.nasa.gov
  Mds-Host-hn=hopper.nas.nasa.gov
  Mds-Host-hn=chapman.nas.nasa.gov
```

Attribute	Value
Mds-Service-protocol	0.1
Mds-validfrom	200301222039.51Z
Mds-validto	200301222040.21Z
Mds-keepto	200301222040.21Z
Mds-Service-port	2119
Mds-Os-release	6.5
Mds-Os-name	IRIX64
objectClass	Mds
objectClass	MdsComputer
objectClass	MdsOs
objectClass	MdsService
objectClass	MdsServiceGram
objectClass	MdsSoftware
Mds-Software-deployment	jobmanager
Mds-Computer-isa	mips
Mds-Service-Gram-schedulertype	pbs
Mds-Computer-manufacturer	sgi
Mds-Service-hn	steger.nas.nasa.gov
Mds-Service-type	x-gram
Mds-Service-url	x-gram://steger.nas.nasa.gov:2119/jobmanager:/O=Grid/O=National Aero

LDAP Browser – Simple Search



Search DN: Mds-Vo-name=nas, o=nasa, o=grid

Filter: (objectclass=MdsHost)

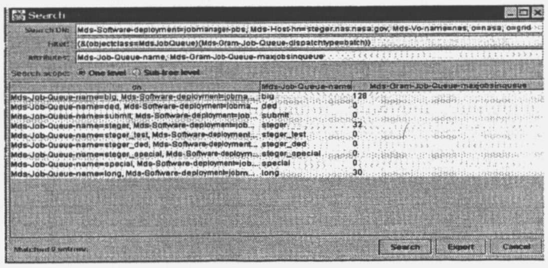
Attributes: Mds-Cpu-Total-count

Search scope: ○ One level ● Sub-tree level

dn	Mds-Cpu-Total-count
Mds-Host-hn=turing.nas.nasa.gov, Mds-Vo-name=nas, o=nasa, o=grid	32
Mds-Host-hn=steger.nas.nasa.gov, Mds-Vo-name=nas, o=nasa, o=grid	128
Mds-Host-hn=chapman.nas.nasa.gov, Mds-Vo-name=nas, o=nasa, o=grid	1024
Mds-Host-hn=lomax.nas.nasa.gov, Mds-Vo-name=nas, o=nasa, o=grid	512
Mds-Host-hn=hopper.nas.nasa.gov, Mds-Vo-name=nas, o=nasa, o=grid	64

Matched 5 entries.

LDAP Browser – Search Filter (contd.)



Search DN: Mds-Software-deployment=jobmanager-pbs, Mds-Host-hn=steger.nas.nasa.gov, Mds-Vo-name=nas, o=nasa, o=grid

Filter: (&(objectclass=MdsJobQueue)(Mds-Gram-Job-Queue-dispatchtype=batch))

Attributes: Mds-Job-Queue-name, Mds-Gram-Job-Queue-maxjobsinqueue

Search scope: ● One level ○ Sub-tree level

dn	Mds-Job-Queue-name	Mds-Gram-Job-Queue-maxjobsinqueue
Mds-Job-Queue-name=big, Mds-Software-deployment=jobma...	big	128
Mds-Job-Queue-name=ded, Mds-Software-deployment=jobma...	ded	0
Mds-Job-Queue-name=submit, Mds-Software-deployment=job...	submit	0
Mds-Job-Queue-name=steger, Mds-Software-deployment=job...	steger	32
Mds-Job-Queue-name=steger_test, Mds-Software-deployment...	steger_test	0
Mds-Job-Queue-name=steger_ded, Mds-Software-deployment...	steger_ded	0
Mds-Job-Queue-name=steger_special, Mds-Software-deploym...	steger_special	0
Mds-Job-Queue-name=special, Mds-Software-deployment=job...	special	0
Mds-Job-Queue-name=long, Mds-Software-deployment=jobm...	long	30

Matched 9 entries.

Launch Pad Resource Page

- The IPG Web Portal, Launch Pad, has a resource page that shows the various resources available on the Grid
- Demo Launch Pad Resource Page
- Launch Pad is accessible from the IPG Home page, or at:

`https://portal.ipg.nasa.gov/launchpad/servlet/launchpad`

*What's Next

- Ok,
  - Now I can use all the basic grid tools to
    - Learn about resources
    - Write scripts to package the grid tools for my application
- Is there anything that makes this a bit easier?
  - Yes,
    - IPG has developed higher level services that help you with some of the items you have already learned
      - IPG Job Manager
      - IPG Broker

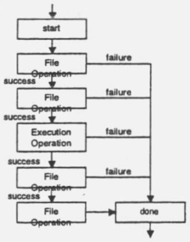
How do they Fit

- Use the Job Manager to:
  - Pre and post stage your data and executables
  - Submit your job(s)
  - Get status on your job(s)
- Use the Broker to:
  - Suggests machines
  - Builds a Job Manager Job and submit it to the Job Manager to run

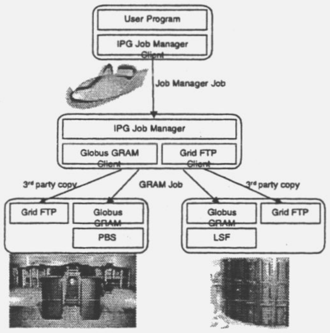
The IPG Job Manager Service

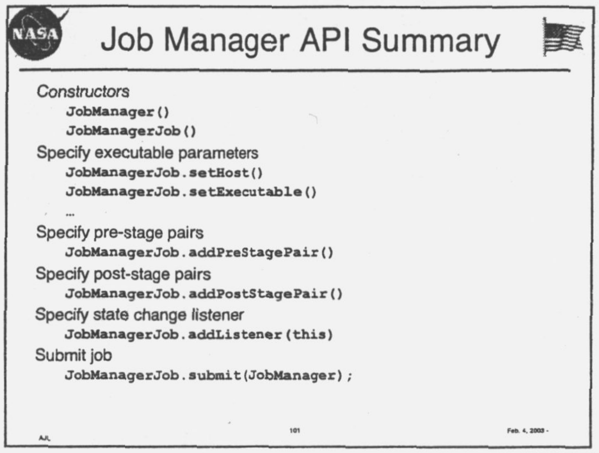
- Reliably execute jobs
- Job is
  - Set of files to pre-stage
    - Source and dest hosts, files and paths, recursive
  - Executable to run
    - Host name, executable on host, arguments, environment, ...
  - Set of files to post-stage
- Maintain a job database
  - Information about jobs maintained for a user-specified amount of time
  - User-specified job information and state changes per job
- Client can be notified of job state changes
- Invocation
  - Java, C++, Perl APIs
  - Launch Pad

Executing a Job Manager Job



IPG Job Manager Architecture





Job Manager API Summary

Constructors

```
JobManager()
JobManagerJob()
```

Specify executable parameters

```
JobManagerJob.setHost()
JobManagerJob.setExecutable()
...
```

Specify pre-stage pairs

```
JobManagerJob.addPreStagePair()
```

Specify post-stage pairs

```
JobManagerJob.addPostStagePair()
```

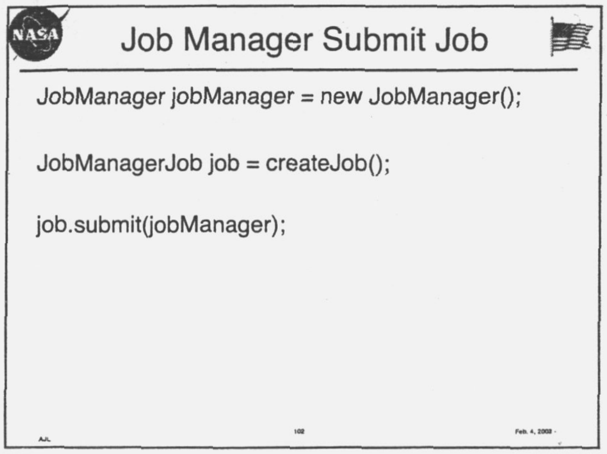
Specify state change listener

```
JobManagerJob.addListener(this)
```

Submit job

```
JobManagerJob.submit(JobManager);
```

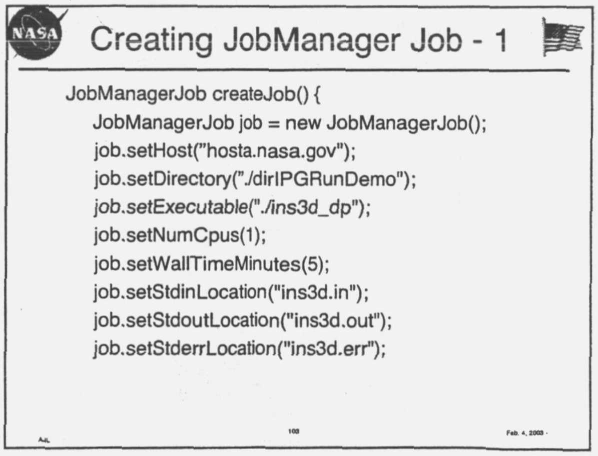
AJL 101 Feb. 4, 2003 -



Job Manager Submit Job

```
JobManager jobManager = new JobManager();

JobManagerJob job = createJob();

job.submit(jobManager);
```

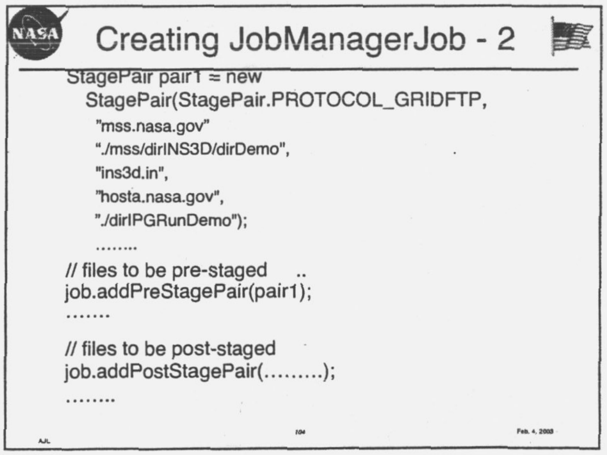
AJL 102 Feb. 4, 2003 -



Creating JobManager Job - 1

```
JobManagerJob createJob() {
    JobManagerJob job = new JobManagerJob();
    job.setHost("hosta.nasa.gov");
    job.setDirectory("./dirIPGRunDemo");
    job.setExecutable("./ins3d_dp");
    job.setNumCpus(1);
    job.setWallTimeMinutes(5);
    job.setStdinLocation("ins3d.in");
    job.setStdoutLocation("ins3d.out");
    job.setStderrLocation("ins3d.err");
```

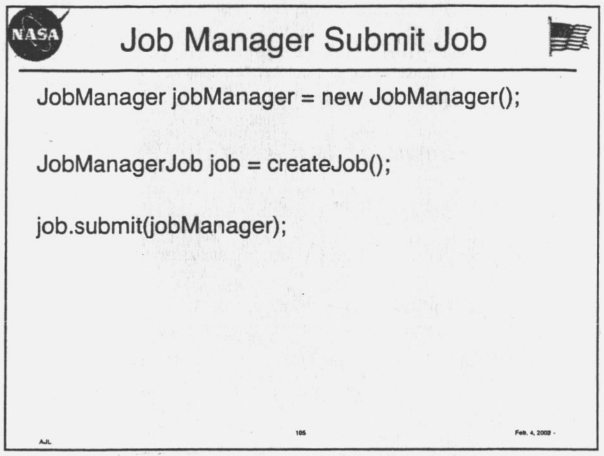
AJL 103 Feb. 4, 2003 -



Creating JobManagerJob - 2

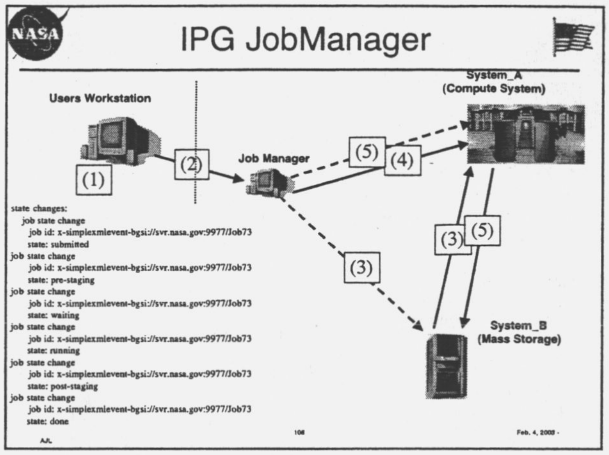
```
StagePair pair1 = new
   StagePair(StagePair.PROTOCOL_GRIDFTP,
      "mss.nasa.gov"
      "./mss/dirINS3D/dirDemo",
      "ins3d.in",
      "hosta.nasa.gov",
      "./dirIPGRunDemo");
      ........
// files to be pre-staged     ..
job.addPreStagePair(pair1);
.......

// files to be post-staged
job.addPostStagePair(.........);
........
```

AJL 104 Feb. 4, 2003



Job Manager Submit Job

```
JobManager jobManager = new JobManager();

JobManagerJob job = createJob();

job.submit(jobManager);
```



IPG JobManager

Users Workstation

(1)

(2)

Job Manager

(5) (4)

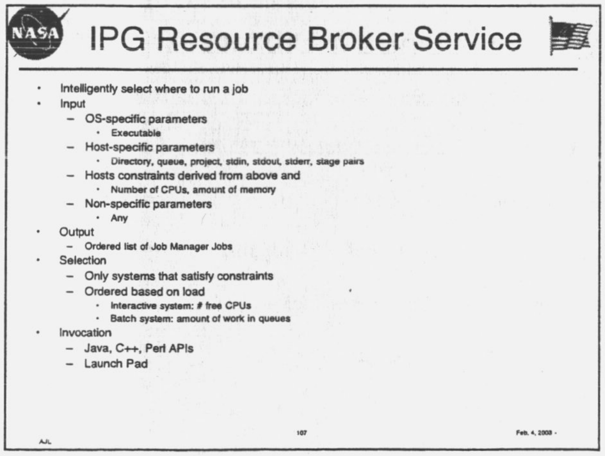
System_A
(Compute System)

(3)

(3) (5)

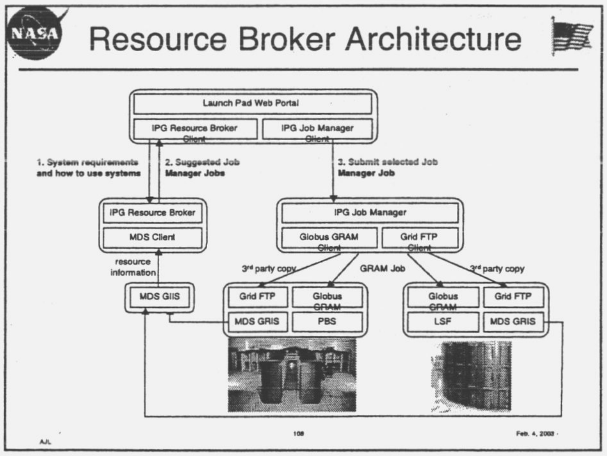
System_B
(Mass Storage)

```
state changes:
   job state change
      job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
      state: submitted
job state change
      job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
      state: pre-staging
job state change
      job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
      state: waiting
job state change
      job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
      state: running
job state change
      job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
      state: post-staging
job state change
      job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
      state: done
```



IPG Resource Broker Service

- Intelligently select where to run a job
- Input
  - OS-specific parameters
    - Executable
  - Host-specific parameters
    - Directory, queue, project, stdin, stdout, stderr, stage pairs
  - Hosts constraints derived from above and
    - Number of CPUs, amount of memory
  - Non-specific parameters
    - Any
- Output
  - Ordered list of Job Manager Jobs
- Selection
  - Only systems that satisfy constraints
  - Ordered based on load
    - Interactive system: # free CPUs
    - Batch system: amount of work in queues
- Invocation
  - Java, C++, Perl APIs
  - Launch Pad



Resource Broker Architecture

Launch Pad Web Portal

IPG Resource Broker Client

IPG Job Manager Client

1. System requirements and how to use systems

2. Suggested Job Manager Jobs

3. Submit selected Job Manager Job

IPG Resource Broker

MDS Client

resource information

MDS GIIS

IPG Job Manager

Globus GRAM Client

Grid FTP Client

3rd party copy

GRAM Job

3rd party copy

Grid FTP

Globus GRAM

MDS GRIS

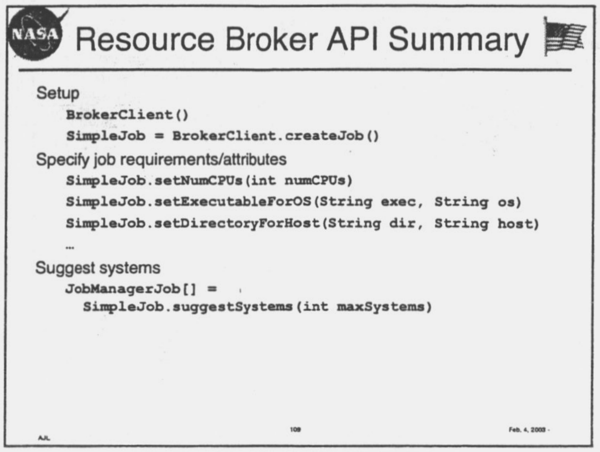
PBS

Globus GRAM

Grid FTP

LSF

MDS GRIS



Resource Broker API Summary

Setup

```
BrokerClient()
SimpleJob = BrokerClient.createJob()
```

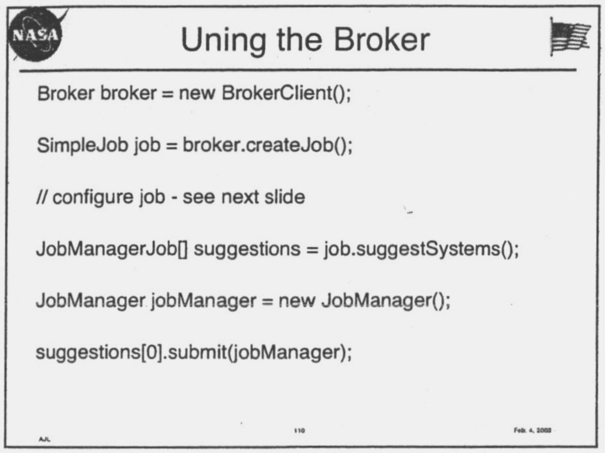
Specify job requirements/attributes

```
SimpleJob.setNumCPUs(int numCPUs)
SimpleJob.setExecutableForOS(String exec, String os)
SimpleJob.setDirectoryForHost(String dir, String host)
...
```

Suggest systems

```
JobManagerJob[] =
  SimpleJob.suggestSystems(int maxSystems)
```

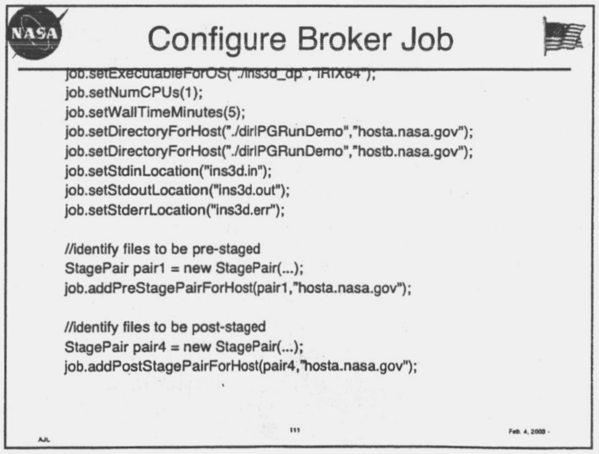
AJL 109 Feb. 4, 2003 -



Uning the Broker

```
Broker broker = new BrokerClient();

SimpleJob job = broker.createJob();

// configure job - see next slide

JobManagerJob[] suggestions = job.suggestSystems();

JobManager jobManager = new JobManager();

suggestions[0].submit(jobManager);
```

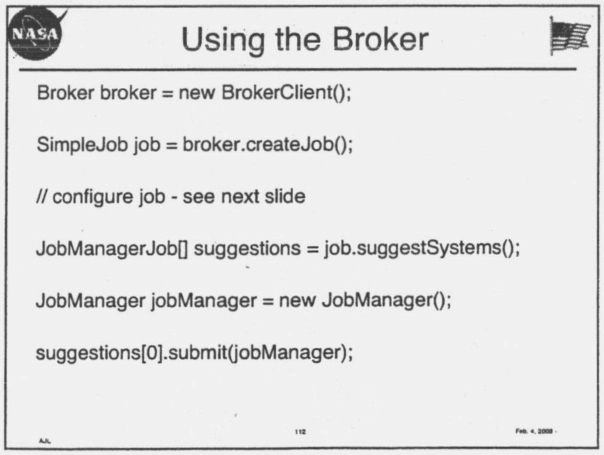
AJL 110 Feb. 4, 2003



Configure Broker Job

```
job.setExecutableForOS("./ins3d_dp","IRIX64");
job.setNumCPUs(1);
job.setWallTimeMinutes(5);
job.setDirectoryForHost("./dirIPGRunDemo","hosta.nasa.gov");
job.setDirectoryForHost("./dirIPGRunDemo","hostb.nasa.gov");
job.setStdinLocation("ins3d.in");
job.setStdoutLocation("ins3d.out");
job.setStderrLocation("ins3d.err");

//identify files to be pre-staged
StagePair pair1 = new StagePair(...);
job.addPreStagePairForHost(pair1,"hosta.nasa.gov");

//identify files to be post-staged
StagePair pair4 = new StagePair(...);
job.addPostStagePairForHost(pair4,"hosta.nasa.gov");
```

AJL 111 Feb. 4, 2003 -

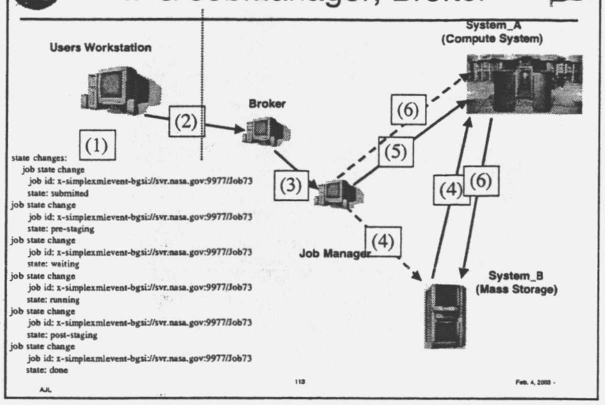


Using the Broker

```
Broker broker = new BrokerClient();

SimpleJob job = broker.createJob();

// configure job - see next slide

JobManagerJob[] suggestions = job.suggestSystems();

JobManager jobManager = new JobManager();

suggestions[0].submit(jobManager);
```

AJL 112 Feb. 4, 2003 -

IPG JobManager, Broker



```
state changes:
  job state change
    job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
    state: submitted
job state change
    job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
    state: pre-staging
job state change
    job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
    state: waiting
job state change
    job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
    state: running
job state change
    job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
    state: post-staging
job state change
    job id: x-simplexmlevent-bgsi://svr.nasa.gov:9977/Job73
    state: done
```

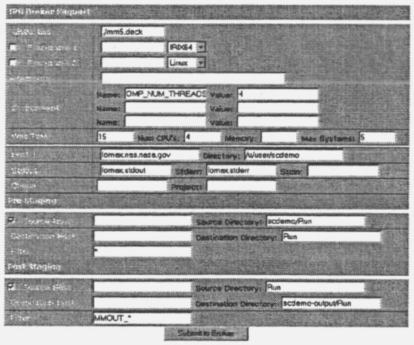
What's Next?

- Ok,
  - Now I think I am pretty knowledgeable about the Grid environment, but I have a question?
  - Earlier, you talked about MyProxy, but we have not used it yet for my application.
- What gives?
  - Well, as we mentioned, MyProxy is for web portals to have a way to access your proxy to execute commands on your behalf.
  - We will now show you a portal IPG has build and how you could use it for you application if you preferred a web interface.

IPG Web Portal

- Launch Pad is the IPG Web Portal
- You can do everything we have demonstrated and more in Launch Pad
- Security is provided using a proxy the same way it is from any system on the grid.
- The proxy is retrieve from a server called MyProxy that stores proxies for this specific use.

Launch Pad Step 1: Enter Parameters



[illegible]: /mm5.deck
□ [illegible]: IRIX64
□ [illegible]: Linux
Name: OMP_NUM_THREADS Value: 4
Name: Value:
Name: Value:
[illegible]: 15 Num CPU's: 4 Memory: Max Systems: 5
[illegible]: lomax.nss.nasa.gov Directory: /u/user/scdemo
[illegible]: lomax.stdout Stderr: lomax.stderr Stdin:
[illegible]: Project:

Pre Staging

☑ [illegible]: Source Directory: scdemo/Run
[illegible]: Destination Directory: Run
[illegible]: *

Post Staging

☑ [illegible]: Source Directory: Run
[illegible]: Destination Directory: scdemo-output/Run
[illegible]: MMOUT_*

Launch Pad Step 2: Select System

IPG BROKER RESULTS

Host Name:lomax.nas.nasa.gov
Executable: /mm5 deck
Arguments:
Directory:/u/user/scdemo/
Environment:
name = OMP_NUM_THREADS
value = 4
Stdout:lomax.stdout
Stderr:lomax.stderr
Stdin:
Number of CPU's:4
Wall time:15
Memory Allocated:
Click here to select

Host Name:steger.nas.nasa.gov
Executable: /mm5 deck
Arguments:
Directory:/u/user/scdemo/
Environment:
name = OMP_NUM_THREADS
value = 4
Stdout:steger.stdout
Stderr:steger.stderr
Stdin:
Number of CPU's:4
Wall time:15
Memory Allocated:
Click here to select

Pre Staging

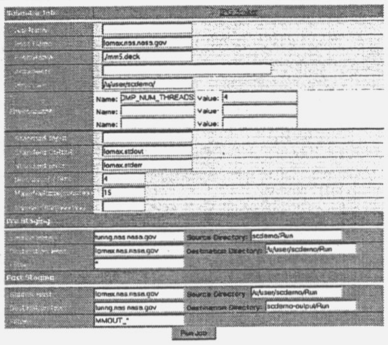
Source Host:turing.nas.nasa.gov **Source Directory:**scdemo/Run
Destination Host:selected host **Destination Directory:**Run
Filter:*

Post Staging

Source Host:selected host **Source Directory:**Run
Destination Host:turing.nas.nasa.gov **Destination Directory:**scdemo-output/Run
Filter:MMOUT_*

AJL 117 Feb. 4, 2003 -

Launch Pad Step 3: Submit Job



[illegible]	IPG Broker		
[illegible]			
[illegible]	lomax.nas.nasa.gov		
[illegible]	./mm5.deck		
[illegible]			
[illegible]	/u/user/scdemo/		
[illegible]	Name: OMP_NUM_THREADS	Value: 4	
	Name:	Value:	
	Name:	Value:	
[illegible]			
[illegible]	lomax.stdout		
[illegible]	lomax.stderr		
[illegible]	4		
[illegible]	15		
[illegible]			
Pre Staging			
[illegible]	turing.nas.nasa.gov	Source Directory:	scdemo/Run
[illegible]	lomax.nas.nasa.gov	Destination Directory:	/u/user/scdemo/Run
[illegible]	*		
Post Staging			
[illegible]	lomax.nas.nasa.gov	Source Directory:	/u/user/scdemo/Run
[illegible]	turing.nas.nasa.gov	Destination Directory:	scdemo-output/Run
[illegible]	MMOUT_*		

AJL 118 Feb. 4, 2003 -

Outline

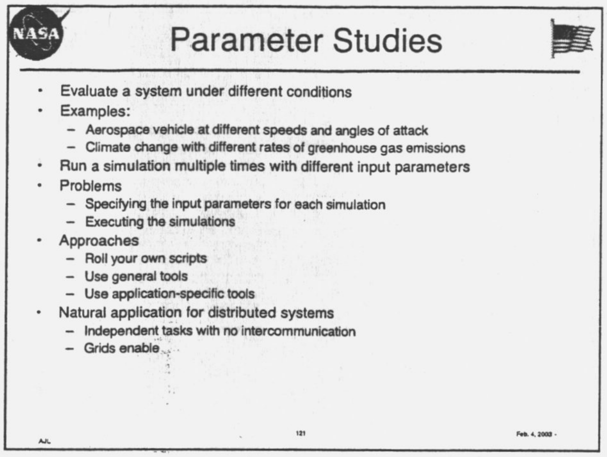
- Introduction
  - What are Grids?
  - Current State of IPG
  - Overview of Grid Middleware
  - Future Directions
- Using the Grid
  - Prerequisites
  - Using Basic Components of the Grid
  - Accessing Grid Services from an application
- Programming Models In Use
  - Parameter Studies
  - Multi-Disciplinary
  - Data-Mining
- Data Management With the SRB

AJL 119 Feb. 4, 2003 -

Programming Models

- Parameter Studies
- Multi-system
- Multi-disciplinary
- Data Mining - Agent Technology

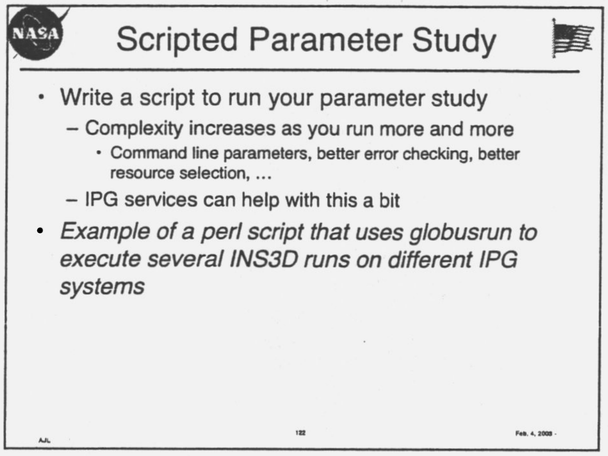
AJL 120 Feb. 4, 2003



Parameter Studies

- Evaluate a system under different conditions
- Examples:
  - Aerospace vehicle at different speeds and angles of attack
  - Climate change with different rates of greenhouse gas emissions
- Run a simulation multiple times with different input parameters
- Problems
  - Specifying the input parameters for each simulation
  - Executing the simulations
- Approaches
  - Roll your own scripts
  - Use general tools
  - Use application-specific tools
- Natural application for distributed systems
  - Independent tasks with no intercommunication
  - Grids enable

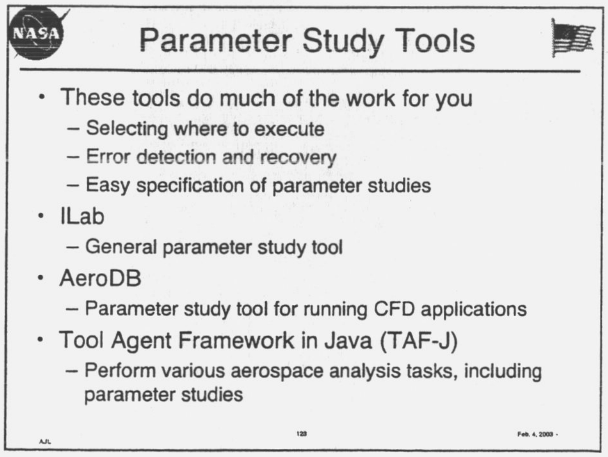
AJL 121 Feb. 4, 2003 -



Scripted Parameter Study

- Write a script to run your parameter study
  - Complexity increases as you run more and more
    - Command line parameters, better error checking, better resource selection, ...
  - IPG services can help with this a bit
- *Example of a perl script that uses globusrun to execute several INS3D runs on different IPG systems*

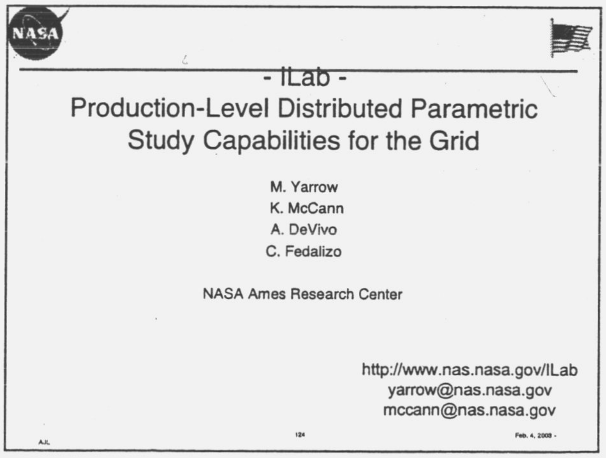
AJL 122 Feb. 4, 2003 -



Parameter Study Tools

- These tools do much of the work for you
  - Selecting where to execute
  - Error detection and recovery
  - Easy specification of parameter studies
- ILab
  - General parameter study tool
- AeroDB
  - Parameter study tool for running CFD applications
- Tool Agent Framework in Java (TAF-J)
  - Perform various aerospace analysis tasks, including parameter studies

AJL 123 Feb. 4, 2003 -



- ILab -
Production-Level Distributed Parametric Study Capabilities for the Grid

M. Yarrow
K. McCann
A. DeVivo
C. Fedalizo

NASA Ames Research Center

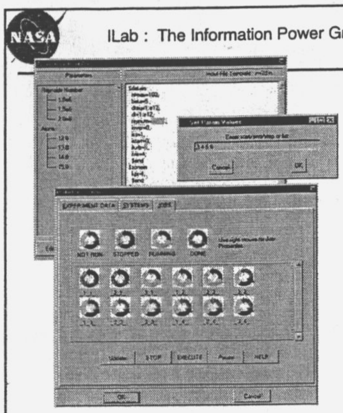
http://www.nas.nasa.gov/ILab
yarrow@nas.nasa.gov
mccann@nas.nasa.gov

AJL 124 Feb. 4, 2003 -

ILab Features

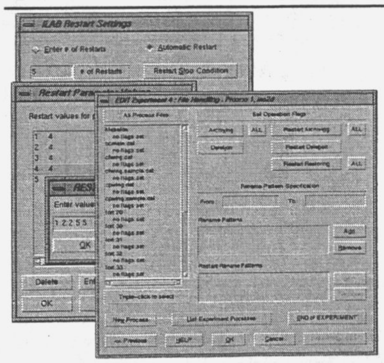
- Automated high throughput on multiple systems
- Operates with Globus (or without - using scp/ssh)
- Transparent access to parallel schedulers and libraries
- Monitoring, logging, and archiving
- Ease of use: sophisticated GUI with built-in HELP; arbitrary parameterization with custom parameterizing editor
- API for programming access (without GUI)
- Restart capability for CFD usage
- Handles complex multistage processes
- Reduces design cycle time - by a factor of up to 100!
- Simple and rapid out-of-box experience
- Proven track record of success and user satisfaction:
  - Users at Boeing (CFD applications, Nastran)
  - Used at UC Davis for CFD studies
  - NASA Ames users (Semiconductor Device Modeling, Rotorcraft blade modeling)
  - Industry (Circuit Simulation)

ILab : The Information Power Grid Virtual Laboratory



- Parameterization made simple and easy
- ILab is really a code generator: generates and submits scripts to execute parameterized jobs
- Absolutely no programming or scripting required
- Several Job Models, including Globus, PBS, LSF, Condor, MPI, local or remote, "ganged"
- Real-time monitoring of job status, job stopping, masking
- Organization and archiving of all "Experiment" data
- MRU and complete "history" secretarial Features
- Built with Object-Oriented programming model

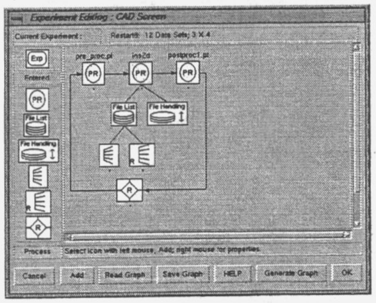
Restart Capability



These functionalities are essential for restarts:

- Settings for number of restarts, I.e., job phases
- Settings for ramped solver parameters
- Archiving, restore specification
- Generalized rename ability between restarts, e.g.: q.*.* ◊ q.*.restart

ILab CAD Process Specification



- Implemented with directed-graph technology
- Allows user specification of arbitrary processes via "drag-and-drop"
- Seen here specifying restart loop, parameterization, and file handling
- Currently limited to supported job models

Getting a Copy of ILab

- See our web site :

  www.nas.nasa.gov/ILab

- Contact :
  yarrow@nas.nasa.gov
  mccann@nas.nasa.gov

- NOTE : NDA required; SUA in process.

AeroDB

Stuart Rogers, Michael J. Aftosmis,
Shishir Pandya, Neal Chaderjian

NASA Advanced Supercomputing Division
NASA Ames Research Center

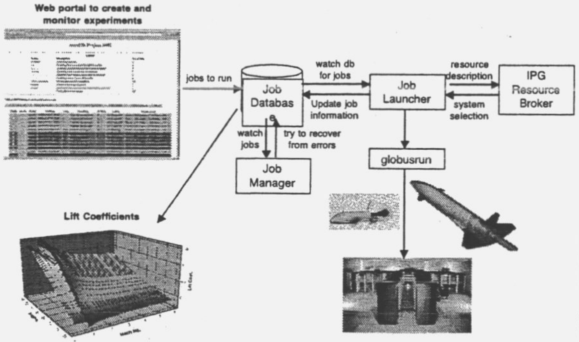
Edward Tejnil, Jasim Ahmad

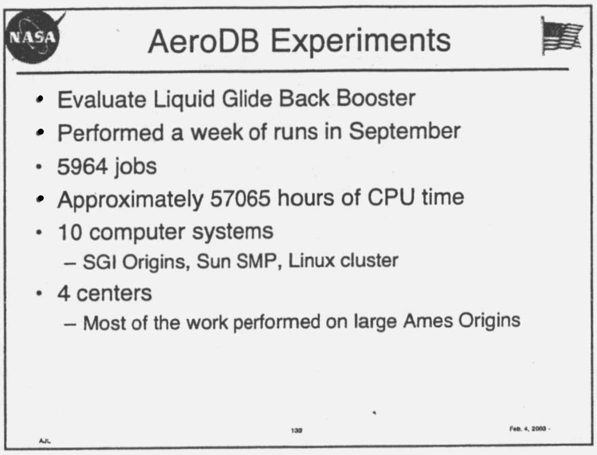
Eloret Institute
NASA Ames Research Center

AeroDB

- Application-specific parameter study tool
- Study performance of an aerospace vehicle under various conditions
  - Angles of attack, side slip angle, Mach number, control surface deflections
  - Thousands of CFD cases
- Web portal to specify and execute
- Little user intervention
- Run a large CFD parameter study on a Liquid Glide Back Booster configuration

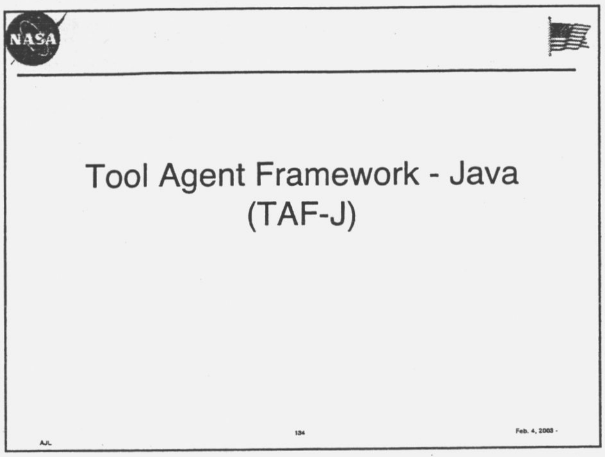
AeroDB Overview



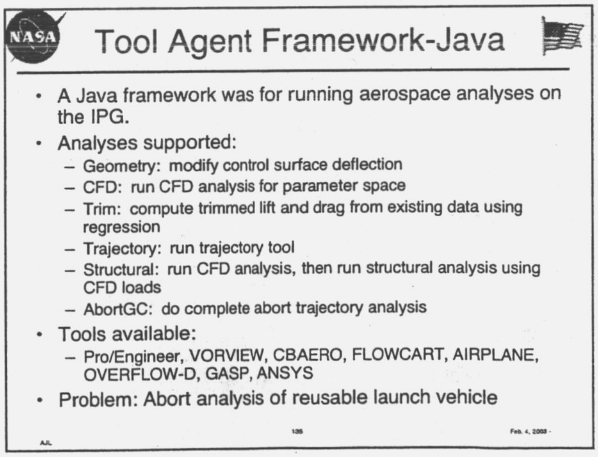


AeroDB Experiments

- Evaluate Liquid Glide Back Booster
- Performed a week of runs in September
- 5964 jobs
- Approximately 57065 hours of CPU time
- 10 computer systems
  - SGI Origins, Sun SMP, Linux cluster
- 4 centers
  - Most of the work performed on large Ames Origins

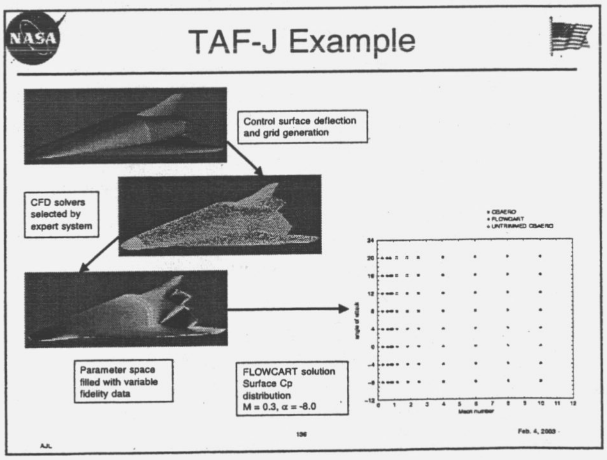


Tool Agent Framework - Java (TAF-J)



Tool Agent Framework-Java

- A Java framework was for running aerospace analyses on the IPG.
- Analyses supported:
  - Geometry: modify control surface deflection
  - CFD: run CFD analysis for parameter space
  - Trim: compute trimmed lift and drag from existing data using regression
  - Trajectory: run trajectory tool
  - Structural: run CFD analysis, then run structural analysis using CFD loads
  - AbortGC: do complete abort trajectory analysis
- Tools available:
  - Pro/Engineer, VORVIEW, CBAERO, FLOWCART, AIRPLANE, OVERFLOW-D, GASP, ANSYS
- Problem: Abort analysis of reusable launch vehicle



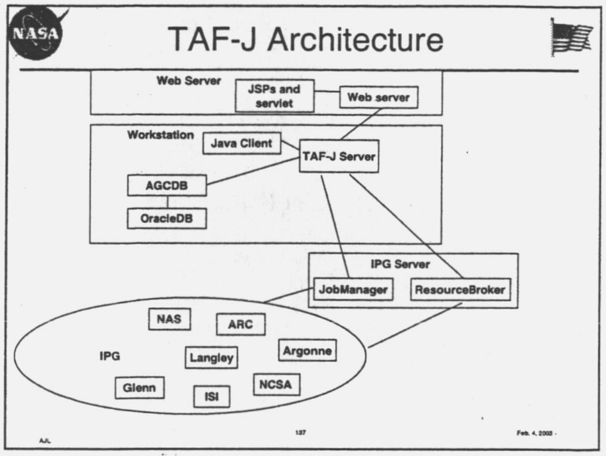
TAF-J Example

Control surface deflection and grid generation

CFD solvers selected by expert system

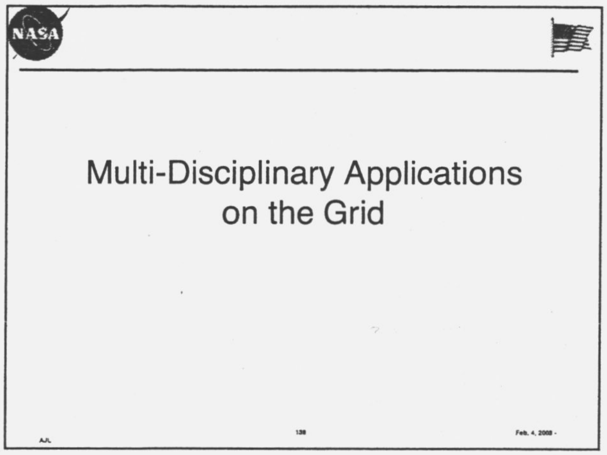
Parameter space filled with variable fidelity data

FLOWCART solution Surface Cp distribution M = 0.3, α = -8.0



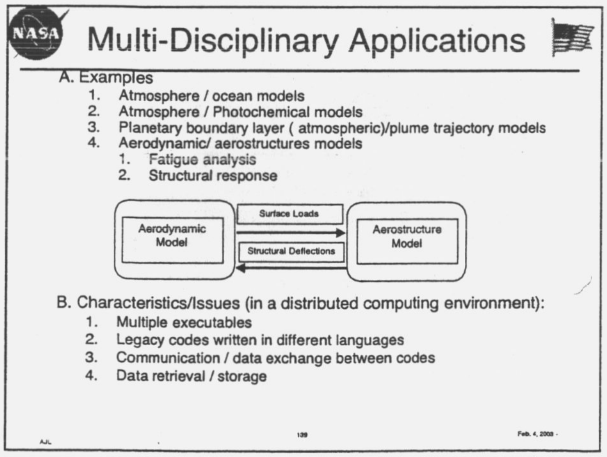
TAF-J Architecture

AJL 137 Feb. 4, 2003 -



Multi-Disciplinary Applications on the Grid

AJL 138 Feb. 4, 2003 -



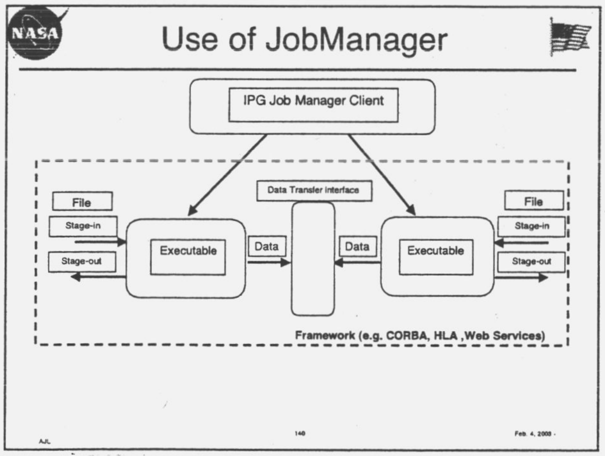
Multi-Disciplinary Applications

A. Examples
1. Atmosphere / ocean models
2. Atmosphere / Photochemical models
3. Planetary boundary layer (atmospheric)/plume trajectory models
4. Aerodynamic/ aerostructures models
    1. Fatigue analysis
    2. Structural response

Aerodynamic Model
Surface Loads
Structural Deflections
Aerostructure Model

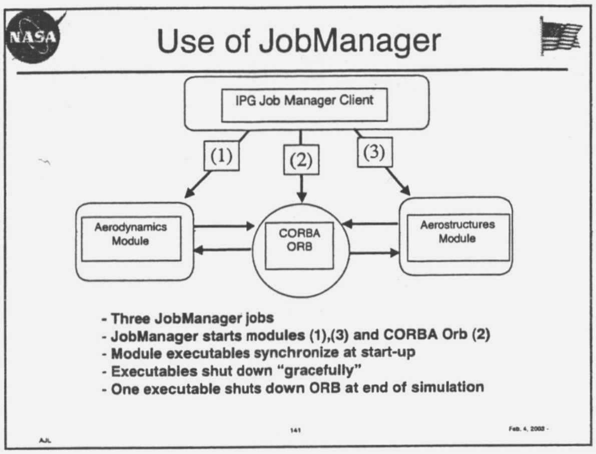
B. Characteristics/Issues (in a distributed computing environment):
1. Multiple executables
2. Legacy codes written in different languages
3. Communication / data exchange between codes
4. Data retrieval / storage

AJL 139 Feb. 4, 2003 -



Use of JobManager

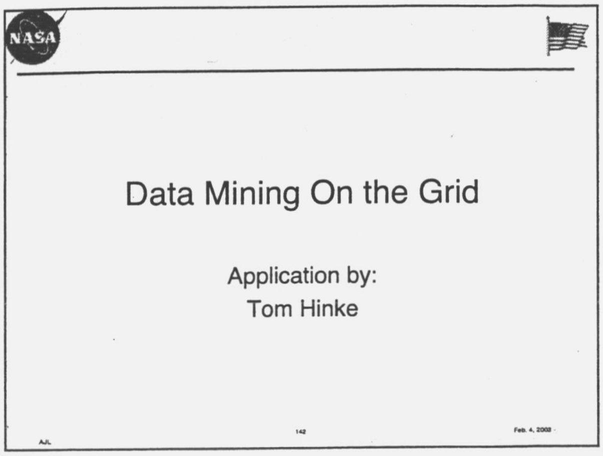
AJL 140 Feb. 4, 2003 -



Use of JobManager

- Three JobManager jobs
- JobManager starts modules (1),(3) and CORBA Orb (2)
- Module executables synchronize at start-up
- Executables shut down "gracefully"
- One executable shuts down ORB at end of simulation

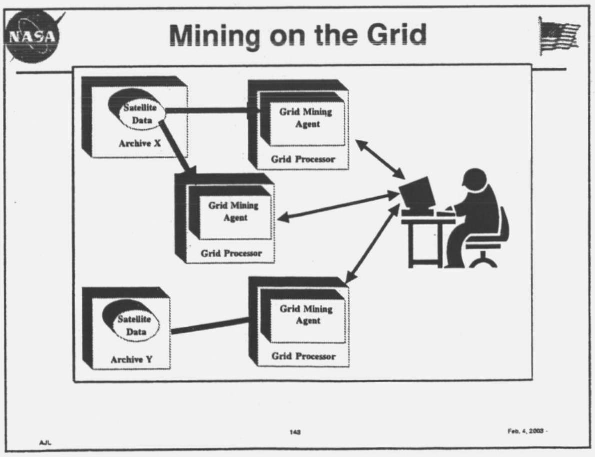
AJL 141 Feb. 4, 2003



Data Mining On the Grid

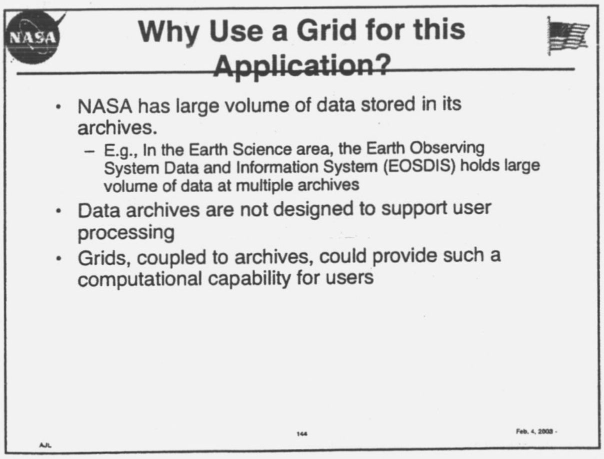
Application by:
Tom Hinke

AJL 142 Feb. 4, 2003



Mining on the Grid

AJL 143 Feb. 4, 2003



Why Use a Grid for this Application?

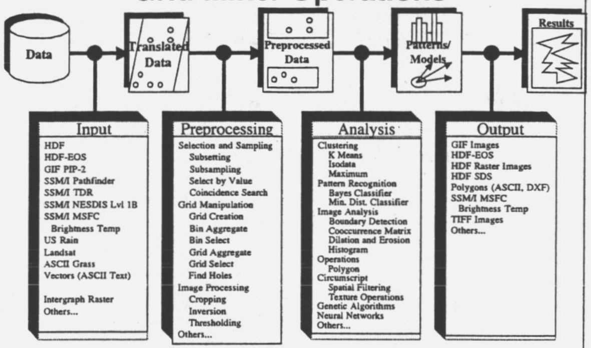
- NASA has large volume of data stored in its archives.
  - E.g., In the Earth Science area, the Earth Observing System Data and Information System (EOSDIS) holds large volume of data at multiple archives
- Data archives are not designed to support user processing
- Grids, coupled to archives, could provide such a computational capability for users

AJL 144 Feb. 4, 2003

Grid Miner

- Developed as one of the early applications on the IPG
  - Helped debug the IPG
  - Provided basis for satisfying one of two major IPG milestones last year
- Provides basis for what could be an on-going Grid Mining Service

Grid Miner Operations



Input
HDF
HDF-EOS
GIF PIP-2
SSM/I Pathfinder
SSM/I TDR
SSM/I NESDIS Lvl 1B
SSM/I MSFC
Brightness Temp
US Rain
Landsat
ASCII Grass
Vectors (ASCII Text)

Intergraph Raster
Others...

Preprocessing
Selection and Sampling
Subsetting
Subsampling
Select by Value
Coincidence Search
Grid Manipulation
Grid Creation
Bin Aggregate
Bin Select
Grid Aggregate
Grid Select
Find Holes
Image Processing
Cropping
Inversion
Thresholding
Others...

Analysis
Clustering
K Means
Isodata
Maximum
Pattern Recognition
Bayes Classifier
Min. Dist. Classifier
Image Analysis
Boundary Detection
Cooccurrence Matrix
Dilation and Erosion
Histogram
Operations
Polygon
Circumscript
Spatial Filtering
Texture Operations
Genetic Algorithms
Neural Networks
Others...

Output
GIF Images
HDF-EOS
HDF Raster Images
HDF SDS
Polygons (ASCII, DXF)
SSM/I MSFC
Brightness Temp
TIFF Images
Others...

Figure thanks to Information and Technology Laboratory at the University of Alabama in Huntsville

Starting Point for Grid Miner

- Grid Miner reused code from object-oriented ADaM data mining system
  - Developed under NASA grant at the University of Alabama in Huntsville
  - Implemented in C++ as stand-alone, objected-oriented mining system
    - Runs on NT, IRIX, Linux
  - Has been used to support research personnel at the Global Hydrology and Climate Center and a few other sites.
- Object-oriented nature of ADaM provided excellent base for enhancements to transform ADaM into Grid Miner

Transforming Stand-Alone Data Miner into Grid Miner

- Original stand-alone miner had 459 C++ classes.
- Had to make small modifications to 5 classes and added 3 new classes
- Grid commands added for
  - Staging miner agent to remote sites
  - Moving data to mining processor

Staging Data Mining Agent to Remote Processor

- globusrun -w -r target_processor '&(executable=$(GLOBUSRUN_GASS_URL)#path_to_agent)(arguments=arg1 arg2 ... argN)(minMemory=500)'

Moving data to be mined

- gsincftpget remote_processor local_directory remote_file

Example: Mining for Mesoscale Convective Systems

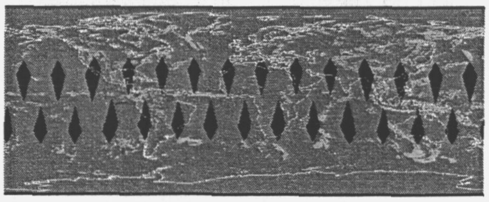


Image shows results from mining SSM/I data

Example of Data Being Mined

- 75 MB for one day of global data - Special Sensor Microwave/Imager (SSM/I).
- Much higher resolution data exists with significantly higher volume.

How to Use SRB on IPG

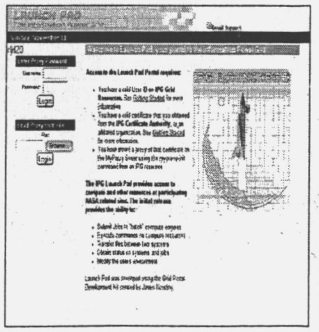
- How to define Metadata
- Commands to move data in and out of jobs
- API to incorporate calls within our sample programs

Summary

- What we have talked about today
- How do I get Help
  - Describe IPG Support Model
  - IPG Web Pages, references

IPG Launch Pad

- Launch Pad
  - A general web portal to access IPG resources and services
    - Start/monitor/stop applications, move files, information about resources, ...
  - In the future, launch pad code can be used as a base to build application-specific portals
- "MyGRID"
  - Project to maintain user specific information that is accessible throughout the grid either via a portal such as LaunchPad or via the command line



Access to the Launch Pad Portal requires:

- You have a valid User ID on IPG Grid Resources. See Getting Started for more information
- You have a valid certificate that you obtained from the IPG Certificate Authority, or an affiliated organization. See Getting Started for more information.
- You have stored a proxy of that certificate on the MyProxy Server using the myproxy-init command from an IPG resource

The IPG Launch Pad provides access to compute and other resources at participating NASA related sites. The initial release provides the ability to:

- Submit Jobs to "batch" compute engines
- Execute commands on compute resources
- Transfer files between two systems
- Obtain status on systems and jobs
- Modify the users environment

Launch Pad was developed using the Grid Portal Development Kit created by Jason Novotny.

Launch Pad Interface